Effectiveness of SAFe methodology on communication and collaboration

Hannele Räsänen

MSc Thesis
Helsinki 14.9.2020

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta/Osasto – Fakultet/Sektion –Faculty/Section	Koulutusohjelma – Studieprogram – Study Programme
Faculty of science	Master's Programme in Computer Science

Tekijä – Författare – Author

Hannele Räsänen

Työn nimi – Arbetets titel – Title

Effectiveness of SAFe methodology on communication and collaboration

Ohjaajat – Handledare – Supervisors

Lea Kutvonen and Tomi Männistö

Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages
MSc Thesis	14.9.2020	67 pages + 18 attachment pages

Tiivistelmä – Referat – Abstract

Nowadays, with the influence of global economy large corporations use global software development to utilise advantages of geographically decentralised organisations and global outsourced software development. Through distributed organisations the work can be done around the clock. Global software development is impacted by three distance dimensions: time distance, geographical distance, and socio-cultural distance, which all bring some challenges. At the same time agile way of working has become more and more popular method in software development. As agile practises are created for co-located teams there is a demand for having working online solutions for communication and collaboration in distributed teams. Corporations use scaled agile way of working to support software development of large initiatives and projects. Scaled Agile Framework (SAFe) is the most popular among the scaled agile methods.

This thesis was conducted as a case study in a multinational corporation. Objective of the case study was to research effectiveness of scaled agile methodology SAFe on communication and collaboration in teams and agile release trains. The case study included two parts: a web-survey and interviews. The results of the analyses of the case study support findings from the literature in the field. The results indicate the importance of communication and collaboration in agile practices and the significance of the online tools that support it.

ACM Computing Classification System (CCS):

•Software and its engineering ~Software creation and management ~Software development process management ~Software development methods ~Agile software development

•Social and professional topics ~Professional topics ~Management of computing and information systems ~Project and people management ~Project management techniques

•Human-centered computing ~Collaborative and social computing ~Collaborative and social computing systems and tools ~Social networking sites

Avainsanat – Nyckelord – Keywords

software development, agile, scaled agile, SAFe, distributed teams, communication, collaboration

Säilytyspaikka – Förvaringställe – Where deposited

University of Helsinki library

Muita tietoja – Övriga uppgifter – Additional information

Software systems

Contents

1 Introduction

With the contribution of globalisation, large corporations have moved to use globally distributed software development model (Ågerfalk, Fitzgerald 2006). The model brings corporations several benefits including cost savings (Holmstrom, Conchuir et al. 2006) and an opportunity to utilize expertise from several countries (Korkala, Abrahamsson 2007). At the same time using agile and scaling agile methods in software development is coming more and more popular worldwide (Razzak, Richardson et al. 2018), (Gustavsson 2018), (Robinson, Sharp 2010).

Well working communication and collaboration are crucial factors for agile teams and initiatives that use scaled agile way of working (Sharp, Helen, Robinson 2010), (Uludag, Kleehaus et al. 2019.), (Kropp, Meier 2016). In distributed agile teams the importance of good communication and well-functioning collaboration is even higher (Ågerfalk, Fitzgerald 2006), (Holmstrom, Conchuir et al. 2006), (Korkala, Abrahamsson 2007). When spread to several locations, the team can no longer interact face-to-face, but instead will have to rely on online communication solutions and collaboration tools (Sharp, Helen, Robinson 2008). Communication and collaboration technologies have evolved in recent years and they offer several communication platforms and collaboration tools, which can be used to support work in globally distributed teams and organisations. Communication platforms include Skype for Business and Microsoft Teams, and collaboration tools GitHub and JIRA (Calefato, Ebert 2019).

Agile manifesto (Beck, K., Beedle et al. 2001) is the origin for agile way of working. The Agile manifesto published in 2001 presented agile values and principles, which are the foundation for agile methods like Kanban, XP and Scrum. Each of these introduce their own values, artifacts and principles that support agile way of working (Robinson, Sharp 2010), (Paasivaara, M., Durasiewicz et al. 2009). At the same time in early 2000's, software development started also to use Lean thinking methodology. Lean methodology (Poppendieck, Poppendieck 2003, pp. 1-10), (Sayer, Williams 2007, pp. 7-46) was developed by Toyota in 1940's to simplify and optimise processes.

Agile methods were developed to be used in small, co-located teams that can interact physically every day (Uludag, Kleehaus et al. 2019.). In large projects several teams

work toward the common goal. As development projects are large in multinational corporations, it brings a demand to scale agile up from team level to govern the whole project. Several scaling agile methodologies has been developed and Scaled Agile Framework (SAFe) is currently the most popular of them (VersionOne Inc. 2019).

Scaled Agile Framework (SAFe) (Uludag, Kleehaus et al. 2019.), (Neve, Godbole et al. 2017) is based on agile and lean methods and its first version was published in 2011. SAFe 4.6 version has four configurations that can be implemented, including essential SAFe, large solution SAFe, portfolio SAFe and full SAFe. Essential SAFe is suitable for smaller companies and full SAFe for large enterprises. Each configuration offers levels to support the different organisation layers' needs. SAFe levels include team, program, solution and portfolio levels.

This thesis offers insight into how Scaled Agile Framework (SAFe) is used in a multinational company that has globally distributed organisation. A case study was conducted in multinational company that is called in this thesis 'Company A'. The case study had two data gathering methods a web-survey and interviews. The focus of the case study is to investigate does the SAFe methodology have an effect on communication and collaboration in agile teams and between teams in agile release trains. The literature in the field suggests that agile and SAFe events and artifacts support information flow, collaboration, and transparency in the teams and between the teams (Putta, Paasivaara et al. 2018a).

Through a web-survey and interviews it was established an overall view on what events are held in 'Company A's' execution structures, and how the respondents perceive them to effect communication and collaboration. An extensive number of questions in the survey supported a possibility to evaluate the situation from many points of view and efforts to discover dependencies and correlation in the circumstances that effected on the responders' answers.

The aim of the case study was to answer three questions a) "How do the practitioners perceive the effects of the events of the SAFe methodology on communication and collaboration?", b) "Do the ways of working of the SAFe methodology add transparency and value to the practitioners?", and c) "What are the correlating factors that effect the experience of the practitioners of the SAFe methodology?".

Chapter 2 provides a detailed description of the values and practices of agile and lean

methods, and introduces four configurations and four levels of Scaled Agile Framework (SAFe) methodology. Statistics from annual state of agile report on usage of different agile and scaled agile methods are also presented. Chapter 2 introduces distance dimensions in context of globally distributed organisations, and communication and collaboration activities and tools in agile settings.

Chapter 3 presents the case study conducted in 'Company A'. The research objective of the case study was to study the usage of SAFe methodology in real-life setting and analyse its effects on communication and collaboration in agile teams and between the teams in agile release trains. The focus of the case study was on SAFe events and ways of working in 'Company A'. The case study consisted of two data gathering methods a web-survey and interviews. Data gathered from the case study was quite large and gave a lot of inside into the usage of Scaled Agile Framework (SAFe) in 'Company A'.

Chapter 4 introduces the results of the analysis of the case study in 'Company A'. The results are presented through findings on how three major events of SAFe effect communication and collaboration in agile teams and agile release trains. The focus of the analysis will be in a daily stand-up meeting, program increment (PI) planning and system demo events. A deeper look will also be taken into iteration events. Additional findings about ways of working, training and using agile coach are also presented.

Chapter 5 contains conclusions in a form of reflections from the result of the case study. Chapter 5 also offers improvement suggestions for discussion, and recommendations for future study.

2 Communication and collaboration in agile set-ups

Nowadays software development of large corporation is often distributed globally on several locations. As the agile teams and team members physically work in different locations, it brings complexity to communication and collaboration within the team and between teams (Ågerfalk, Fitzgerald 2006), (Holmstrom, Conchuir et al. 2006), (Korkala, Abrahamsson 2007).

The agile software development methods like Scrum, XP, Kanban and Lean support the agile way of working in organisations in a team level (Dikert, Paasivaara et al. 2016). Scaled Agile Framework (SAFe) methodology has been developed

to support managing large-scale agile projects and to transform the whole organisation agile (Putta, Paasivaara et al. 2018a).

2.1 Communication and collaboration

There are several definitions for *communication*. One of the ways to define communication is used in Business English: "*the process of sharing information, especially when this increases understanding between people or groups*"[1]. A vast number of communication theories exist (Littlejohn, Foss 2009), starting from the *Classical Rhetorical Theory* of the ancient Roman philosophers (Littlejohn, Foss 2009, pp. 103-108) to *Computer-Mediated Communication* (Littlejohn, Foss 2009, pp. 161-164). The classical rhetorical theory is seen as a foundation for the modern field of communication and the computer-mediated communication theory contains any method of communication aided by the digital technology.

The Sapir-Whorf Hypothesis (Littlejohn, Foss 2009, p. 867) says: "*Communicative behaviour includes spoken words, written messages, nonverbal signals, and our reactions to all of these*". Communication can be divided into few main categories 1) verbal communication, 2) written form of communication, 3) nonverbal communication, and 4) visual communication, which all have several sub-categories.

Verbal communication (Littlejohn, Foss 2009, p. 864) can be seen as a set of rules in verbal behaviour. Examples of these rules include sounds that can be said to mean something and meaning of the words through context.

Written communication (Littlejohn, Foss 2009, p. 934) is communication through the written word. It can take forms such as early written storytelling, mathematics formulas (Littlejohn, Foss 2009, p. 657), and modern ways of electronic communications including email and text messages (Littlejohn, Foss 2009, p. 2).

Nonverbal communication (Littlejohn, Foss 2009, pp. 690-694) was studied first by Charles Darwin in 1872. Nowadays nonverbal communication is a part of studies in several different fields including social psychology and linguistics. Gestures like body language, facial expressions, tone of the voice, and touch are signs in nonverbal communication. Nonverbal communication reveals people's inner feeling through their eyes, the way they walk, stand, and cross their arms.

[1] Cambridge University dictionary. https://dictionary.cambridge.org/dictionary/english/communication, [19.9.2019]

Visual communication (Littlejohn, Foss 2009, pp. 1002-1005) is something that can be seen like art, photographs, technical graphics and moving pictures. The importance of visual communication has increased in 21st century due to all the time growing development of visual technologies. These days the stream of images that people see every day through different medias is uncountable. Almost everything that can be perceived visually can be categorised as signs of visual communication.

Collaboration can be defined as people working together toward common goal or purpose[2] (Tabaka 2006, p. 3). Collaboration happens between individuals, teams, companies (Aira 2012) or even countries. Most of the work nowadays includes some kind of collaboration with others (Aira 2012), (Mistrik, Grundy et al. 2010, pp. vii-xiii). Collaboration can vary from a) people working together on solving challenges, b) giving one's own expertise in a team to accomplish goals, and c) to countries working together on inhibiting global warming. Even team sports would not be very successful without collaboration.

With today's geographically decentralised organisations (Ågerfalk, Fitzgerald 2006) the collaboration isn't anymore happening only face-to-face, as organisations are distributed over cities, countries, and even continents. Decentralisation brings a demand for new kind of *Collaboration Technologies* (Hummel, Rosenkranz et al. 2016), (Paasivaara, M., Durasiewicz et al. 2009) that supports the collaboration over geographical distance (Ågerfalk, Fitzgerald 2006), (Holmstrom, Conchuir et al. 2006), (Ågerfalk, Fitzgerald et al. 2005). Collaboration tools like Skype for Business, Microsoft Teams, email, phone, desktop sharing and videoconference (Hummel, Rosenkranz et al. 2016), (Paasivaara, M., Durasiewicz et al. 2009) are commonly used collaboration methods in distributed organisations, as well as in people's personal lives. Remote working is also becoming more and more popular, which means that even people working in the same physical location don't see each other every work day (Sharp, Helen, Robinson 2008).

2.2 *Agile methods and practices, and Lean principles*

Agile software development is becoming one of the most commonly used methods for software development. Agile methods are based on Agile manifesto, that introduced in 2001 four agile values and twelve principles (Beck, K., Beedle et al. 2001). Agile way of working uses principles and values, which are shared by many agile methodologies,

[2] Cambridge University dictionary. https://dictionary.cambridge.org/dictionary/english/collaboration, [15.9.2019]

processes, and practices[3]. The most common agile methods are *Scrum, XP, Kanban* and hybrids of the methods (VersionOne Inc. 2019). Software development also started to adapt principles from *Lean* methodology in early 2000's. Lean methodology's purpose is to get maximum value for the customer with minimum waste. Lean aims to optimisation and continuous improvement of processes, effort and resources (Poppendieck, Poppendieck 2003, pp. 1-10), (Sayer, Williams 2007, pp. 7-46).

2.2.1 Kanban

Kanban (Gross, McInnis 2003) is a Japanese word that means signboard. Kanban board was invented by Taiichi Onho when he was working for Toyota, to reduce costs and minimize the work in process (WIP[4]). Kanban is used for scheduling work and visualising status of the work in a one sight.

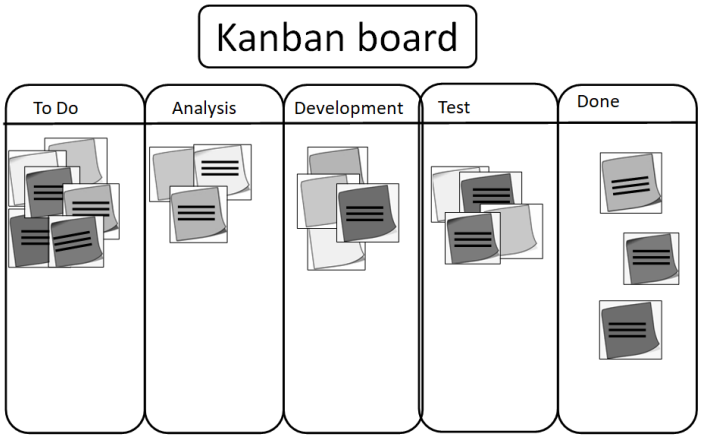


Figure 2.1: Kanban board. Figure is based on Kanban board (Leopold, Kaltenecker 2015)

There are variation on the realisation and naming of the lanes on the Kanban board. One example of lanes of a Kanban board is 1) To Do, 2) Analysis, 3) Development, 4) Test, and 5) Done (Leopold, Kaltenecker 2015). Work items are parked on the Kanban board according to the stage that they are in 1) when the work item is chosen to be done next, it is placed into the 'To Do' -lane, 2) the work item will move to the 'Analysis' -lane when it is ready for further processing, 3) when requirements for the work item are ready, it is moved to the 'Development' -lane, 4) it is further to be moved to 'test' –lane, when it is ready for testing, and 5) finally to be moved to 'Done' –lane, when it is ready for production. See Figure 2.1 for illustration of Kanban board lanes.

[3] Scrum Alliance®, Inc, Scrum Theory. https://www.scrumalliance.org/about-scrum/theory, [9.1.2020]
[4] Kanbanize, 2020. What is a Kanban WIP Limit? https://kanbanize.com/kanban-resources/getting-started/what-is-wip/, [9.1.2020]

Kanban has six core principles[5][6] (Leopold, Kaltenecker 2015) including make work visible, implement feedback mechanisms, and improve collaboratively practices. Detailed descriptions of Kanban core principles can be found in Table 2.1.

Kanban core principles	Description
Make work visible	*Make work visible* is about visualisation of what the team is doing. Visualisation is happening through the Kanban board and the lanes in it. Through making the work visual everyone in the team can see at all times, what is the status of the work of the team.
Limit work in progress (WIP)	*Limit work in progress (WIP)* is a number of work items a team is currently working with. It shows the workflow capacity of the team in any given moment. Limit the WIP is used to manage the workflow to be smooth and to prevent overload of the team. WIP limits the number of work items on the lanes of the Kanban board and helps the team to focus on the current tasks and to avoid bottlenecks.
Manage flow	*Manage flow* practice is to optimise as much as possible time-to-market and decrease the lead-time. Lead-time is the time from when the work is started until it is finished. Managing the flow also means that the ongoing work is finished before starting a new one.
Make progress policies explicit	Purpose of *Make progress policies explicit* is to be transparent to all members of the team. All team members know and understand the policies and principles used in the work of the team and follow them. Second purpose is that when the policies and practices are clear they can be improved.
Implement feedback mechanisms	*Implement feedback mechanisms* purpose is to focus in continuous improvement. Learning can happen thought feedback from customers and others outside the team and also through feedback inside the team from the team events like daily stand-up meeting.
Improve collaboratively practices	Purpose of *Improve collaboratively practices* is to always find better ways to work to improve practise Manage lead-time of the flow. That can be done by finding continuous improvement from any new theory or way of working.

Table 2.1: Kanban core principles

2.2.2 Extreme Programming (XP)

Extreme Programming (XP)[7] (Kuppuswami, Vivekanandan et al. 2003) is a technical process model, which determines development work through values and practices. In XP[8] the team work is an essential part of the method. The stakeholders including customer and managers work together with a development team, forming together well collaborating team. XP method is best suited for small software development teams. In XP the team works in iterations and the goal of the iteration is in the end of each one to

[5] Kanbanize, Kanban Explained for Beginners. https://kanbanize.com/kanban-resources/getting-started/what-is-kanban/, [9.1.2020]
[6] Packt Publishing Limited, 2020. The six core practices of Kanban. https://subscription.packtpub.com/book/business/9781783000906/1/ch01lvl1sec04/the-six-core-practices-of-kanban, [9.1.2020]
7 Wells, 1999. The Values of Extreme Programming. http://www.extremeprogramming.org/values.html, [9.1.2020]
[8] Wells, 1999. Extreme Programming: A gentle introduction. http://www.extremeprogramming.org/,[16.5.2020]

have tested stories that are ready to go into production.

XP introduces four core values and twelve practice[9][10] (Kuppuswami, Vivekanandan et al. 2003), (Beck, K. 1999), (Beck, Kent 2000, pp. 29-63). Purpose of the values is to guide the way of working in the team. Core values include communication and simplicity, which emphasise the importance of sharing knowledge and keeping the code simple. See description of all XP values in Table 2.2.

XP values	**Description**
Communication	*Communication* in the team is to share knowledge in face-to-face conversations and to work together with all of tasks of the team from requirements to coding and testing.
Simplicity	*Simplicity* is to keep the code and the design simple and to avoid waste. That is why the purpose of it is only to do what is needed and askes, nothing more.
Feedback	*Feedback* is to constantly improve the performance of the team from iteration to another. Based on feedback from demonstrations, adjustments to the design and implementation of product are made.
Courage	*Courage* is to raise issues when they occur, stop doing things in the way that does not work, and to give realistic estimates.
Respect	*Respect* is to respect all members of the team and to work together toward common goals.

Table 2.2: XP values.

XP practices are based on the XP values and they are designed to work together. The purpose of XP practises is to bring more tangibility on how to use the more theoretical values in practice. XP practices include planning game, small releases, and open workspace. See description of all XP practices in Table 2.3.

XP practices	**Description**
Planning game	*Planning game* is an event held in the beginning of the iteration where the team and the customer together make the plan for the next iteration. Customer decides the prioritisation for the stories and the team gives estimations for them.
Small releases	Aim of *small releases* is to release the first version quickly and to use continuous integration. Through getting feedback and bug detecting, the team can improve the product to the next release.
Metaphor	*Metaphor* is a high level design of the software system, which will guide the team in the development and it is easy to understand also by others.
Simple design	*Simple design* contains only the current requirements, has no duplicate code, and is the simplest version that works.

9 Agile Alliance. Extreme programming. https://www.agilealliance.org/glossary/xp/, [9.1.2020]
10 Wells, 1999. The Values of Extreme Programming. http://www.extremeprogramming.org/values.html, [9.1.2020]

Continuous testing	*Continuous testing* means that unit testing is done continually, so allowing quick feedback from the tests to the coders. Unit tests can be written even before the real code. Functional testing done by the customer can also be run continuously.
Refactoring	Goal of *Refactoring* is to continuously improve quality of the code. This is done by simplifying, adding flexibility and coherency, and by removing duplicates from the code.
Pair programming	*Pair programming* means that all codes are written jointly by two programmers using one computer. Pair programming includes writing the code, reviewing it, and giving improvement suggestions to the code.
Continuous integration	*Continuous integration* means that the new code is integrated and build, as often as possible, even in every few hours.
Collective ownership	*Collective ownership* is that all the developers share the responsibility of the code and when seen necessary they can improve any part of the code in the system.
On-site customer	*On-site customer* means that customer is present and available for the team to answer questions, set priorities to the stories, and settle disagreements.
40-hour week	*40-hour week* is about keeping the work-life balance. 40-hour week enables work to be done well, and team members to be motivated and be at their best.
Open workspace	*Open workspace* means a common open office premises for the whole team.

Table 2.3: XP practices.

XP has two communication artifacts story cards and wall (Sharp, H., Robinson et al. 2006), which are used for communication about the stories done in the team. Team usually has a collaboration area in their common open space, were team members have put up a wall where the story cards are displayed. Story cards (stories) are kind of index cards, which contain information about the work items including description and estimation of a progress status. The wall can contain simple Kanban board lanes and iteration weeks demonstrating when the tasks are expected to be developed. Nowadays there are several electronic tools where the stories can be written and the wall can be built.

2.2.3 Scrum

The most used agile method is *Scrum,* which is a process model for project management (Paasivaara, M., Durasiewicz et al. 2009). Frequent repetition is a foundation of Scrum; it enables fast handling of arising change needs (Pries, Quigley 2010). Scrum approach[11] (Resnick, Bjork et al. 2011, p. 13) is iterative and incremental, with a purpose to optimize predictability and control the risks (Rahman, Mollah et al. 2018). *Three*

[11] ScrumGuides.org, The Scrum Guide™. https://www.scrumguides.org/scrum-guide.html, [8.1.2020]

pillars of Scrum[12] (Rahman, Mollah et al. 2018) include transparency, inspection, and adaptation. The objective of the three pillars is to support execution of the empirical process control, which is the foundation of scrum. See description of the three pillars of scrum in Table 2.4.

Three pillars of Scrum	Description
Transparency	*Transparency* is visibility of the process and state of the product for all those who are responsible. Common standards, language, and 'definition of done' ensure understanding. Scrum review events give transparency to team members and stakeholders.
Inspection	*Inspection* is frequent inspection of artefacts and progress, which is done to keep the sprint objectives on track. Scrum review and retrospective events give opportunities for inspection.
Adaptation	The purpose of *Adaptation* is to detect that the progress is in acceptable limits. If it deviates too much, adjustments need to be made. Scrum events for Inspect and Adapt are Sprint Planning, Daily Scrum, Spring Review, and Sprint Retrospective.

Table 2.4: The three pillars of Scrum.

The three pillars of scrum are supported by *scrum values*[13] (Rahman, Mollah et al. 2018), (Inayat, Salim et al. 2017). Objective of scrum values is to give guidance to the behaviour and activities of scrum team members. There are five scrum values including among others commitment and openness. The scrum values are described in Table 2.5.

Scrum values	Description
Commitment	*Commitment* is an essential part of scrum team and agile culture. Team commits and works together toward common goals. Team takes only tasks they can commit to and the ones they can complete.
Courage	*Courage* is important success factor to a scrum team. Have courage and feel safe to do the right things, ask for help, and work with challenging problems is essential part of being functional agile team.
Focus	*Focus* means that the scrum team finishes what they have started and are aware of the limits of work in progress (WIP).
Openness	*Openness* is about constantly looking for new ideas, ways of working, and opportunities to learn new things. The team is open about the work and its challenges, and asks for help when needed.

[12] Scrum Alliance®, Inc, 2020. Scrum Theory. https://www.scrumalliance.org/about-scrum/theory, [8.1.2020]
[13] Scrum Alliance®, Inc, 2020. Scrum Values. https://www.scrumalliance.org/about-scrum/values, [8.1.2020]

Respect	*Respect* means understanding the importance of collaboration, valuing each other's ideas, and acknowledging each other to be talented and autonomous people.

Table 2.5: Scrum values.

Scrum calls iterations *sprints* (Paasivaara, M., Durasiewicz et al. 2009), (Resnick, Bjork et al. 2011, pp 13-14), (Pries, Quigley 2010, pp. 31-41), (Blankenship, Bussa et al. 2011, pp. 8-14). Sprints are timeframes that can last from one week to a month. The idea of a sprint is to list tasks that the team commits to do in the timeframe of the sprint. The sprints are sets of time periods that happen sequentially, one always following the other.

Scrum artifacts (Paasivaara, M., Durasiewicz et al. 2009), (Pries, Quigley 2010, pp. 31-41), (Resnick, Bjork et al. 2011, pp. 58-64), (Blankenship, Bussa et al. 2011, pp. 17-21) are product backlog, sprint backlog, sprint burndown chart, increment and 'definition of done' (DoD). The purpose of the scrum artifacts is to give transparency and to help manage the work.

Product backlog contains continuously evolving list of all the objects to be developed for the product. The responsibility of the product owner is to maintain the content of the product backlog and priority of the object in there. The product backlog contains features, requirements and fixes. Product owner decides what tasks in the product backlog are done next and moves them in to the sprint backlog.

Sprint backlog contains the selected items from the product backlog that have been committed by the scrum team to be developed in the ongoing sprint. Sprint backlog visualises the work that is needed to achieve the goals of the sprint. Only development team can add into the sprint backlog after the sprint has begun and if there is a need to drop items that needs to be agreed with the product owner.

Sprint Burndown Chart is a way of monitoring the remaining work in the sprint. Burndown chart indicates if the progress is slower than expected and if there is a risk that all tasks in the sprint will not be completed.

'Definition of Done' (DoD) can be declared when all are in agreement and understand what done means. 'Definition of done' can mean different thing in different teams, but the team has to have only one understanding of it. 'Definition of done' helps development team to estimate how many items from product backlog can be committed to be completed in each sprint. 'Definition of done' means tested and approved work items ready to be released.

Increment is the cumulative number of completed tasks until and during current sprint. Increment is the potentially releasable product in the end of the sprint. The increment has to meet the criteria for 'definition of done'.

Scrum team is a high-performing, self-organised, autonomous, and cross-functional team that has inside the team the needed skills to complete the task given to the team (Pries, Quigley 2010). Roles of scrum team (Pries, Quigley 2010, p. 52), (Resnick, Bjork et al. 2011, pp. 25-35), (Blankenship, Bussa et al. 2011, pp. 21-23) include product owner, delivery team and scrum master. The detailed description of the roles can be found in Table 2.6.

Scrum team roles	**Description**
Product owner	*The Product owner* represents the customer and is responsible for getting the best value for the product the team is developing. Product owner prioritises the work of the team, ensures that the items on product backlog are optimised, and that the product backlog is transparent and clear to everyone.
Delivery team	*Delivery team* is autonomic, self-organised team who manages their own work. Team has from five to nine cross-functional professionals who have expert skills needed to deliver items in the product backlog.
Scrum master	*Scrum master* is a servant leader who facilities the scrum team in its activities, removes obstacles, and upholds that scrum practices, rules and values.

Table 2.6: Scrum team roles

Scrum has four *events* (Paasivaara, M., Durasiewicz et al. 2009), (Resnick, Bjork et al. 2011, pp. 65-73), (Blankenship, Bussa et al. 2011, pp. 23-25) that occur in each sprint, daily scrum, sprint planning, sprint review and sprint retrospective. Purpose of these time-boxed events is to 1) get cadence to the sprint, 2) increase transparency, collaboration and clarity, and 3) to reduce need for other meetings.

Sprint planning is held before each sprint starts. In a sprint planning event the scrum team collaborates and discusses what can be delivered in the sprint, the work is prioritised, and the sprint goals are defined. The scrum master facilitates the meeting, the product owner gives objectives and acceptance criteria, and development team decides the capacity of the team. The output of the meeting is the prioritised product backlog.

Daily Scrum is an event that is held every weekday. Daily scrum lasts for 15 minutes and its purpose is to increase communication, collaboration, progress, and performance in the team. In the daily scrum each team member describes how their work is ongoing

with answering to three questions 1) “What have you done since the last meeting?”, 2) “What do you plan to do before next meeting?”, and 3) “Are there any impediment in your way?”. Daily scrum is an opportunity for the development team to inspect and adapt their work every day. A *meet after meeting* can be held immediately after daily scrum. Purpose of the meet after meeting is that the whole team or some members from it can continue discussing topics that arised in the daily scrum.

Sprint review event is held in the end of every sprint. The purpose of a sprint review is to discuss the work that has and has not been completed in the sprint, and the next tasks to be done. The outcome of the spring review event is adjusted and revised product backlog.

Sprint retrospective event inspects how the sprint went and what could be potential areas for improvement. Questions “What went well?”, “What did not go so well?”, and “What can be improved?” are asked in the sprint retrospective meeting to ensure continues improvement. In sprint retrospective team can do quantitative reviews and qualitative reviews. A quantitative review uses agreed metrics that can be velocity related or simple questions like has the goals been met. In a qualitative review the team can evaluate improvement items by focusing in what could be done better in the next sprint.

2.2.4 Lean

Lean (Poppendieck, Poppendieck 2003, pp. 1-10), (Sayer, Williams 2007, pp. 7-46) is based into Just-in-Time production line that was developed in 1940’s by Toyota. Just-in-Time bases on making only the most minimum needed, at the time it is needed, and only the amount that is needed. Lean thinking aims to produce more effective processes by more leaner and simpler processes. In 2000’s Lean -thinking was adapted into use in software development to produce more enhanced processes. Lean uses kanban board that contains all the work items placed in their current stages.

There are seven *principles* of Lean thinking including among others eliminate waste, amplify learning, and empower the team (Poppendieck, Poppendieck 2003, pp xxv-xxvii). Purpose of lean principles[14] is to support the creation of organisational efficiency and effectiveness. Lean principles are descripted in Table 2.7.

[14]The Lean Way 2016-2020. The Five Principles of Lean. https://theleanway.net/The-Five-Principles-of-Lean [24.5.2020]

Lean principles	Description
Eliminate waste	*Eliminate waste* means that everything not giving customer value at the moment is waste. Waste can come in any form: requirements of a system that is not yet developed, code that is not used or products made by a factory that are not sold fast.
Amplify learning	*Amplify learning* means that software development is a learning process. Software development is making variations and versions in an iterative way, and learning from all the challenges and mistakes that have been encountered.
Decide as late as possible	*Decide as late as possible* means waiting until the last minute before locking decision. Late decision is based as much as possible for the known facts and not on speculation.
Deliver as fast as possible	*Deliver as fast as possible* supports faster time-to-market, as it assures that customer gets what they need at this moment, not what they needed earlier, but the delivery was delayed.
Empower the team	*Empower the team* refers to giving decision power to the experts in the team who have the best knowledge of the matter. That motivates people and makes the process faster.
Build integrity in	*Build integrity in* means that the system meets the users' expectations, it is made with good quality, it works as a whole, and that it is flexible enough for a changes needed.
See the whole	*See the whole* means that it is important also to see the system as a whole and not only focus in the parts of it like DW or GUI, as all the parts have to also work together.

Table 2.7: Lean principles.

2.3 Usage of agile methods

The Annual State of Agile survey (VersionOne Inc. 2019) gathers data every year from over 1000 software professionals from all continents. The latest survey was the 13th and the report was published in the May 2019. The statistics from the survey show the most common techniques, tools and benefits of the agile methods.

- Most common *agile methods* used are 1) Scrum 54%, 2) Other Hybrid 14%, 3) Scrum/XP hybrid 10%, 4) Scrumban 8%, 5) Kanban 5%, 6) Lean Startup 2%, and XP 1%.
- Five Top *agile techniques* are 1) Daily stand-up, 2) Sprint / Iteration planning, 3) Retrospectives, 4) Sprint / Iteration Reviews, and 5) Short Iterations. Planning poker, Team estimation, Kanban, and Release Planning being next on the list.
- Top Five of *Engineering practices* is 1) Unit testing, 2) Coding standards, 3) Continuous integration, 4) Refactoring, and 5) Continuous delivery.
- Most used *agile project management tools* are JIRA 65% and Excel 48%.
- Five Top ranking *benefits* on using agile are 1) Ability to manage changing priorities, 2) Project visibility, 3) Business / IT alignment, 4) Team morale, and 5) Delivery speed / time to market.

- To manage *Outsourced* development projects, 46% of respondents use agile practices
- Among the respondents 78% say that they work in *distributed agile team*.

Scrum, XP, Kanban and Lean are methods that are designed to be used in team level of agile and lean development, not with large projects or organisations. As large corporations have also initiatives where big system landscapes or entire technologies are changed, there was an increasing need for an agile method that would support large projects in their development efforts (Paasivaara, M. 2017).

2.4 *Scaled agile Framework (SAFe) methodology*

Leffingwell introduced in 2011 (Stojanov, Turetken et al. 2015), (Razzak, Richardson et al. 2018) a new methodology called *Scaled Agile Framework (SAFe).* SAFe methodology (Putta, Paasivaara et al. 2018b), (Uludag, Kleehaus et al. 2019.), (Putta, Paasivaara et al. 2018a), (Scaled Agile Inc 2016) offers guidance, principles and practices for scaling agile in an enterprise level. SAFe is based on agile and lean values and principles. SAFe provides a way for several agile teams to collaborate, communicate, synchronise alignments and deliver together. In the 13th annual state of agile survey (VersionOne Inc. 2019) SAFe was ranked number one methodology in scaling agile. The newest SAFe version 5.0 came out in the beginning of 2020. This thesis refers to SAFe version 4.6.

SAFe 4.6 has four configurations[15] (Putta, Paasivaara et al. 2018b), (Uludag, Kleehaus et al. 2019.), (Neve, Godbole et al. 2017), (Scaled Agile Inc. 2018) of the framework Essential SAFe, Large Solution SAFe, Portfolio SAFe, and Full SAFe. The four configurations contain from two to four levels including Team level, Program level, Large Solution level and Portfolio level. SAFe levels are presented in Figure 2.2.

Essential SAFe is best suited to development of solution that have small number of agile teams and few independently developed systems. Essential SAFe has basic elements of the framework and it introduces the core competencies of lean-agile leadership, team and technical agility, DevOps, and release on demand. *Large Solution SAFe* is best suited for those who develop large and complex solutions. Large solution configuration supports several agile release trains (ARTs) and suppliers. Large solution SAFe presents business solutions and lean systems engineering.

[15] Scaled agile Inc, Scaled Agile Framework. https://v46.scaledagileframework.com/#, [8.1.2020]

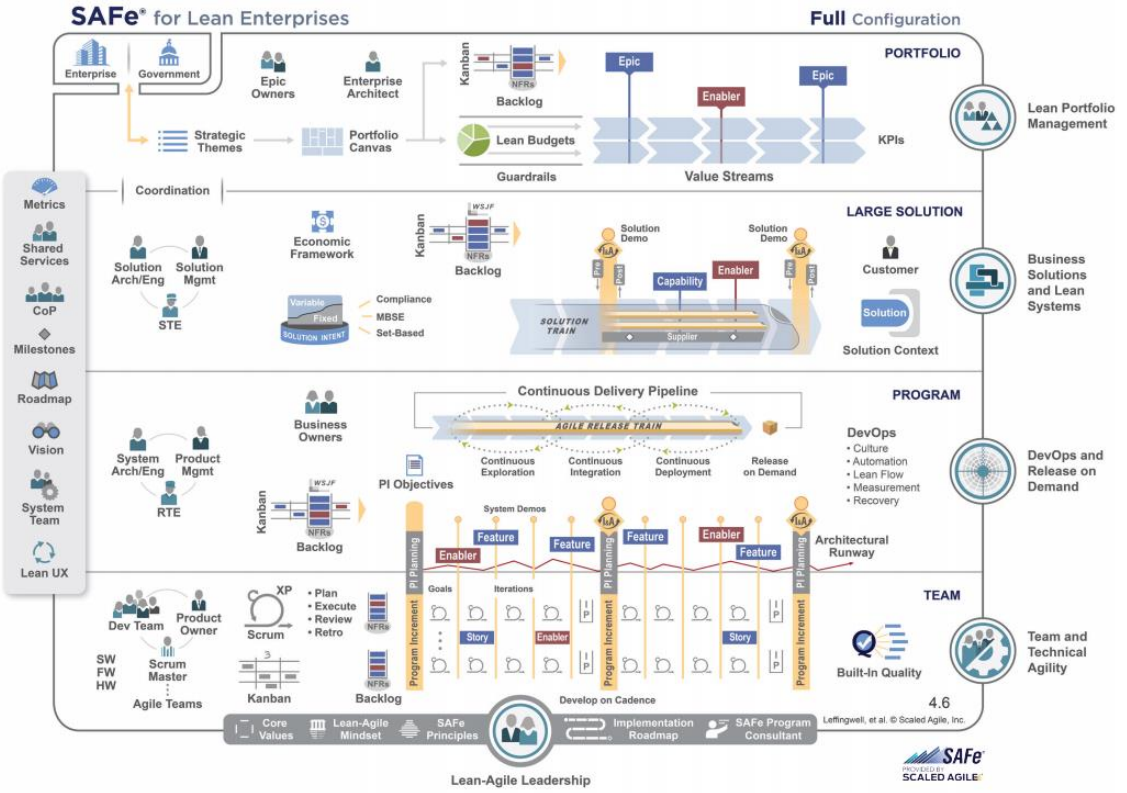


Figure 2.2: Scaled agile Framework version 4.6, Full configuration. © Scaled Agile, Inc.(Scaled Agile 2019)

Portfolio SAFe introduces portfolio strategy and investment funding, agile portfolio operations, and lean governance. Portfolio SAFe uses the lean portfolio management competency that is used to align strategy and execution. *Full SAFe* is the most inclusive configuration and it is best suited for integrated, large scale agile development, which has tens of agile teams and systems landscapes that have many interdependencies.

SAFe has traditional scrum team roles at team level and introduces new roles in program, large solution, and portfolio levels (Putta, Paasivaara et al. 2018a). In addition the program, large solution, and portfolio levels (Razzak, Richardson et al. 2018) introduce 1) new artifacts including epic and capability, 2) program and solution level backlogs, and 3) events including program increment (PI) planning, system and solution demo, scrum of scrum meeting (SoS), and product owner (PO) and ART syncs[16].

The four *core values*[17] (Putta, Paasivaara et al. 2018b) of SAFe are alignment, built-in quality, transparency, and program execution. The SAFe core values guide transformation and operation, and they define behaviour and action in order to determine that development has the right direction to achieve business goals. The four SAFe core values are described in Table 2.8.

[16] Scaled Agile, Inc, Program Increment. https://www.scaledagileframework.com/program-increment/, [11.1.2020]
[17] Scaled Agile, Inc, Core Values. https://v46.scaledagileframework.com/safe-core-values/ , [11.1.2020]

SAFe core values	Description
Alignment	*Alignment* is achieved by clear communication of the strategy and vision. Alignment relies on the enterprise level objectives and that teams and ART yield their own objectives to serve the higher ones. Strategy and vision are communicated through program increment objectives and iteration goals, to express expectations and gain commitment.
Built-in quality	*Built-in quality* is committing to produce quality in every increment, iteration, and in the entire development lifecycle, and to put efforts into maintenance and reducing technical debt.
Transparency	*Transparency* visualises work through program increment objectives, portfolio kanban, and program and team backlogs. Improvements to backlog are created in inspect and adapt event what is aiming for everyone to have understanding of teams and programs WIP. Transparency means to be open, take responsibility of mistakes and to celebrate learnings. For transparency and openness trust is needed. Trust is a key in building high-performance teams and programs.
Program execution	*Program execution* is assumed ability of every team and agile release train to deliver value reliably and efficiently. Leaders as business owners prioritise, reflect, help to adjust scope, and remove impediments to reach goals of the increment.

Table 2.8: SAFe core values

There are nine SAFe Lean-Agile principles[18] (Scaled Agile Inc 2016) which are based on a) lean and agile principles and methods, b) lean product development, c) system thinking, and d) observation of successful enterprises. Purpose of these principles is to inspire and advise about the roles and practices of SAFe and to influence leadership behaviour and decision-making. SAFe principles are described in Table 2.9.

SAFe principles	Description
Take an economic view	*Take an economic view* is to deliver best value and quality in the shortest sustainable lead time. To take an economic view an understanding of economic driven decisions is needed. Trade-offs between risk, cost of delay, operational and development costs has to be defined, and decentralized decision-making has to be supported.
Apply system thinking	*Apply system thinking* is to optimise the full value stream solution itself and the enterprise building it. This is realised through understanding of the larger aim of the system and commitment to the common goals.
Assume variability, preserve options	*Assume variability, preserve options* is to delay decisions to the last reasonable moment and until the last moment explore alternatives for requirements and design options.
Build incrementally with fast,	*Build incrementally with fast, integrated learning cycles* is to develop incremental-

[18] Scaled Agile, Inc., 2019. SAFe Principles. https://v46.scaledagileframework.com/safe-lean-agile-principles/, [11.1.2020]

integrated learning cycles	ly in short iteration cycles and to gain knowledge, evaluate options, and inform decision making with using fast customer feedback, risk mitigation, and prototyping.
Base milestones on objective evaluation of working systems	*Base milestones on objective evaluation of working systems* is about system builders and customers having a shared responsibility for new solutions delivering financial value. Incremental development demonstrates in each iteration point the feasibility of the work in progress.
Visualize and limit WIP, reduce batch sizes, and manage queue length	*Visualize and limit WIP, reduce batch sizes, and manage queue length* is to use small batches of work and to visualise actual capacity, control work in progress (WIP) to assure reliable flow, and to use small queues to reduce wait time.
Apply cadence; synchronize with cross-domain planning	*Apply cadence; synchronize with cross-domain planning* are tools for operating effectively in uncertain development environment. Unpredictable events can be transformed to be predictable with cadence as it provides rhythm to the work.
Unlock the intrinsic motivation of knowledge workers	*Unlock the intrinsic motivation of knowledge workers* is about creating an environment where knowledge workers can have ideas, innovate, and be engaged to the work they do. Engagement comes from autonomy, mission, and purpose.
Decentralize decision-making	*Decentralize decision-making* empowers and motivates teams. Giving autonomy in decision-making to the teams enables reduction of delay, quick decision-making, and faster time-to-market.

Table 2.9: SAFe principles

2.4.1 Team Level

Team level (Paasivaara, M. 2017), (Stojanov, Turetken et al. 2015), (Razzak, Richardson et al. 2018) is the level of agile teams that work in fixed iteration length and timetable. In team level teams use a combination of agile practices of Scrum, XP, and Kanban, and Lean principles. Practices used in the team level include events iteration planning, iteration review, iteration retrospective, backlog refinement, daily stand-up-meetings, and innovation and planning iteration. In team level teams use artifacts like a backlog and stories. In the team level teams are jointly responsible for software definitions, builds, and testing of common solution. Teams usually include from 5 to 10 team members and teams have their own individual iteration backlogs.

Team level includes traditional agile roles like a) a scrum-master who facilitates team events, is a coach to the team, and a driver for agile behaviour, b) a product owner who is the customer, prioritises the work of the team, owns the backlog of the team, and who also defines and accepts requirements, and c) development team that consists of coders, testers, and IT & business analysts who together define, code, and test the developed products and services.

2.4.2 Program Level

Program level (Paasivaara, M. 2017), (Stojanov, Turetken et al. 2015), (Razzak, Richardson et al. 2018), (Alqudah Mashal, Razali Rozilawati 2016) introduces new artifacts, roles, events, and scaled planning activities. Agile release train (ART) designs the strategic vision and roadmap of the program for the teams. Organised under single ART are normally teams that have common goals and delivery dependencies with each other. Program level has its own backlog that contains features within the ART that are delivered for the value stream. Program level introduces three new weekly events including scrum of scrums (SoS), product owner (PO) sync, and ART sync. These events support the execution of the program increment and with the focus into the progress they help to keep the ART on its path. The events are led by the release train engineer (RTE). Scrum masters and product owners participate to events as representatives of their teams. In a Program increment (PI) planning event a program increment plan is made, which contains from eight to twelve weeks of development plans for the whole ART.

Agile Release Train[19] *(ART)* (Paasivaara, M. 2017), (Stojanov, Turetken et al. 2015), (Pries-Heje, Krohn 2017) is a virtual program structure that can be viewed as a team of agile teams. Agile release train is long-term, self –organised execution structure that produces releases in fixed iterations inside an increment. The purpose of the ART is to build solutions that provide value and benefit for the end user. ARTs typically include from 5 to 12 agile teams and it is a virtual organisation for 50 to 125 people. Inside ART teams have common business and technology mission and vision, and the teams have one common program backlog to support the ART. Agile release trains are cross-functional and they have needed capabilities to deliver continuous flow of value; by defining, implementing, testing, deploying and releasing the work they do. ART follows the timetable of program increment and the plans that have been agreed in program increment (PI) planning meeting. ARTs are organised to provide releases for a value stream and it produces a release in the end of each increment, which is a common accomplishment of all the teams in the ART.

Program backlog[20] (Uludag, Kleehaus et al. 2019.) is used through program level kan-

[19] Scaled Agile, Inc., 2019. *Agile Release Train*. https://www.scaledagileframework.com/agile-release-train/, [12.1.2020]
[20] Scaled Agile, Inc. 2018. Program and Solution Backlogs. https://v46.scaledagileframework.com/program-and-solution-backlogs/, [19.1.2020]

ban system, which supports the flow of features and enablers in different states of the backlog. Program backlog consists of the features and enablers of the ART, which fulfil user needs and bring business benefits. Product management is responsible for the program backlog. In collaboration with other stakeholders, product management adds items to the program backlog as a result of the continuous exploration process. The purpose of the *continuous exploration*[21] is to build alignment in the program and inspire innovation.

Feature is a work item that fits into a program increment and it is developed by the ART. All features have a benefit hypothesis that enables the measurement of each feature. Features are prioritised in the program backlog and split into user stories to support development.

Enablers[22] are work items, which purpose is to support and bring visibility to work needed to establish effective development. There are four types of enablers including exploration, architectural, infrastructure and compliance, and they can be found in all of the levels of SAFe.

Program Increment (PI) (Paasivaara, M. 2017), (Razzak, Richardson et al. 2018) is a fixed time period that usually run from eighth to twelve weeks and it consists two-week-long iterations. In the end of each increment ART produces an increment level tested and working release of software or a system. The release is a joint release containing work from all the teams in the ART, who work within the increment.

Program Increment (PI) planning (Uludag, Kleehaus et al. 2019.), (Putta, Paasivaara et al. 2018a) event is a two day planning event that is held in the beginning of each program increment. Planning event gathers all relevant stakeholders and all the teams in the ART together to plan and prioritise the work for the next release. The aim of the planning event is to solve dependencies, create transparency between the teams, and to handle risks. As a result from the planning event, a commitment to agreed objectives for next program increment should be accomplished.

The program increment (PI) planning event has a two-day fixed agenda[23] that includes 1) introductions of business context, 2) solution and architecture visions, 3) planning context, 4) team breakouts time, 5) draft plan review, 6) management review and prob-

[21] Scaled Agile, Inc., 2019. Continuos exploration. https://v46.scaledagileframework.com/continuous-exploration/ ,[23.5.2020]
[22] Scaled Agile, Inc., 2019. Enablers. https://v46.scaledagileframework.com/enablers/[31.8.2020]
[23] Scaled Agile, Inc. 2019. PI Planning. https://www.scaledagileframework.com/pi-planning/, [12.1.2020]

lem solving, 7) planning adjustments, 8) final plan review and risks, 9) confidence vote, and 10) plan rework and planning retrospective.

Preparation is a crucial part of a PI planning event. Preparation includes 1) coordination and communication between product owners, teams and stakeholders, 2) customers to produce a vision and top features to be developed, and 3) a technical vision made by a system architect to support planning. Together these visions and plans give guidance to the delivery of the features of the increment.

In addition to the program increment (PI) planning event the SAFe introduces new events including an ART level system demo and an ART sync meeting for product owners and scrum masters. The program level events are described in Table 2.10.

Program level events	**Description**
Innovation and planning iteration	*Innovation and planning iteration* is a part of each program increment and it gives an opportunity for all teams to work with activities that do not fit into other iterations. The purpose of an Innovation and planning iteration is to dedicate time for innovation, exploration, and continuous learning in the program. An innovation and planning iteration includes events like PI planning, system demo, inspect and adapt workshop and backlog refinement.
System demo	In *System demo* each team from the ART presents their achievements. Purpose of the demo is to give and receive feedback from other teams, business owner, and other stakeholders on the progress and the value of features demoed in the system demo.
Scrum-of-scrum (SoS)	*Scrum-of-scrum (SoS)* is a program level event for coordinating dependencies and bringing transparency to the progress and impediments of the ART. A release train engineer acts as a facilitator of the meeting and all scrum masters are to attend. Other members from the development team participate in SoS when needed.
Product Owner syncs	*Product Owner syncs* is a meeting for POs and it is facilitated by product manager or RTE. Product owner sync follows the progress of the ART and in there product owners discuss problems, opportunities, and possible needs for scope adjustments.
ART Sync meeting	ARTs can also combine the scrum-of-scrum meeting and product owner sync into a one common *ART Sync meeting*.

Table 2.10 Program level events.

Program level also introduces new roles to drive Lean governance through coordination of ARTs and providing alignment under a shared mission. Program level roles include release train engineer (RTE), who can be seen as a chief scrum master and business owner, who has the program level responsibility of the initiative. Program level roles are described in Table 2.11.

Program level roles	Description
Release Train Engineer (RTE)	*Release Train Engineer (RTE)* acts as a chief scrum-master for the train and facilitates ART level meetings including PI planning meeting, SoS, PO and ART syncs.
Product management	*Product management* owns, defines and prioritises backlog in the program level.
S*ystem architect*	S*ystem architect* provides guidance in architectural and technical issues in the program level.
Business owners	Responsibilities of the *Business owner* include governance, compliance, and responsibility for the return of interest.
The *system team*	The *system team* supports with infrastructure and integration, and perform ART-level testing
DevOps	*DevOps* is responsible for building the deployment and automation, and cooperation between teams and the operations department.

Table 2.11: Program level roles.

2.4.3 Large Solution Level

Large Solution[24] (Scaled Agile Inc. 2018) is a level that can be used when developed solutions are demanding and complex. Large solution level is used when there is a need for multiple synchronised agile release trains working together to deliver large and complex systems. Large Solution level introduces *Solutions train*, which synchronises and integrates work of several ARTs. *Solution train engineer* (STE) facilities and guides the efforts of all the ARTs in the solution train.

Solution demo[25] within a large solution level presents joint results from all the ARTs in the solution train. In a solution demo the teams present 1) new capabilities in the solution, 2) compliancy of the solution with Non Functional Requirements (NRF), and 3) that the solution still fits its intendent purpose, so rationalising the continuation of its development. Solution demo is an opportunity for evaluation and feedback, but also for celebrating accomplishments.

Solution backlog[26] is used through solution kanban system, which supports the flow of features and capabilities in the different states of backlog including funnel and analysing. Solution backlog consists of capabilities and enablers of several ARTs, which makes it possible for the development of the whole solution to progress. Solution management is responsible for solution backlog. As a result from continuous exploration

[24] Scaled Agile, Inc., 2019. Large Solution Level. https://v46.scaledagileframework.com/large-solution-level/, [12.1.2020]
[25] Scaled Agile, Inc., 2018. Solution Demo. https://v46.scaledagileframework.com/solution-demo/, [18.1.2020]
[26] Scaled Agile Inc. 2018. Program and Solution Backlogs. https://v46.scaledagileframework.com/program-and-solution-backlogs/, [19.1.2020]

process, which is done in collaboration with other stakeholders, solution management add items to the solution backlog.

Capability is a work item that extents over several ARTs and consists several features. Capabilities fulfil stakeholder needs and are maintained in the backlog of solution train. Even though the work of a capability is spread over several ARTs it is sized to be developed during one program increment.

Large solution level has three additional roles. These roles provide solution level governance and coordination for the multiple ARTs and suppliers of the solution train. The three roles include a) *Solution train engineer (STE),* who facilitates and guides all ARTs and suppliers in the solution train, b) S*olution management,* who has authority for the content in large solution level, and c) S*olution architect,* who is the representative for a team that makes the common technical and architectural definition for the solution.

2.4.4 Portfolio Level

Portfolio level[27] (Paasivaara, M. 2017), (Stojanov, Turetken et al. 2015), (Razzak, Richardson et al. 2018) is used by large businesses that often have multiple agile release trains and development value streams. Portfolios are connected to business strategic themes and lean budgeting of business lines. In Portfolio level large initiatives are called *epics* (Alqudah Mashal, Razali Rozilawati 2016), which are business and architectural changes that run through the value streams. The portfolio level provides budgeting, governance, and coordination to multiple value streams and large development initiatives. Portfolio level introduces also new artifacts including the portfolio canvas and portfolio backlog.

Epic represents solution development initiatives. They are located in program, portfolio and large solutions levels. There are two types of epics: business epics that deliver business value and supporting enabler epics. Epics are split into capabilities and features, which both are needed to gain value for the solution through definition, planning, and implementation. Epics, capabilities, features, and stories can be seen as a hierarchy of work needed to be done. Epics are divided into smaller parts through capabilities and features until they become small enough work item to fit into stories that can be implemented in a single iteration.

[27] Scaled Agile Inc. 2018. Portfolio Level. https://v46.scaledagileframework.com/portfolio-level/, [12.1.2020]

Portfolio canvas defines and describes the structure and purpose of the SAFe portfolios. Portfolio canvas defines SAFe portfolios: value propositions and solutions of the value stream that are delivered to their customer and the key people. Portfolio canvas streamlines portfolio planning and development, facilitates communication in the team, and aligns objectives of the portfolio.

Portfolio Kanban gives transparency to epics and with support from WIP thinking it limits the demand to match the capacity in the value streams and ARTs. With portfolio kanban epics flow on backlogs is managed, made visual, and their prioritisation is analysed. The prioritised and analysed epics are released after their development approval.

Portfolio backlog is the highest level backlog of SAFe and it contains the business and enabler epics. After the epics are approved in portfolio kanban they are released to ARTs and solution trains to kick of the implementation.

In portfolio level three new roles are introduced. These roles are accountable and they govern the highest level of SAFe, as their coordination responsibility spreads over multiple value streams. The new roles include a) *Epic owner*, who coordinates work across value streams in a portfolio, b) *Enterprise architect*, who works across value streams in the portfolio providing strategic technical vision, and c) *Lean portfolio management* (LPM), who has the decision-making power and is financially accountable for the portfolio level.

2.5 Usage of scaling agile methods

The Annual State of Agile survey (VersionOne Inc. 2019) also gathers data about scaling agile.

- Most common *scaling agile methods and approaches* are 1) Scaled Agile Framework (SAFe) 30%, 2) Scrum of Scrums 16%, 3) internally created methods 8%, 4) Disciplined Agile Delivery (DAD) 8%, 5) Large Scale Scrum (LeSS) 3%, and 6) Lean management 3%.
- Top Five *Tips for Success* with scaling agile are 1) internal agile coaches, 2) executive sponsorship, 3) company provided training programs, 4) consistent practices and processes across teams, and 5) implementation of a common tool across teams.

- Top Six *Challenges* in adopting and scaling agile are 1) organizational culture at odds with agile values, 2) general organization resistance to change, 3) inadequate management support and sponsorship, 4) lack of skills / experience with agile methods, 5) inconsistent processes and practices across teams, and 6) insufficient training and education.

2.6 Distributed organisations and distance dimensions

Global Software Development (Paasivaara, M., Durasiewicz et al. 2009) has created new opportunities for an enterprise to implement software development. Organisations in corporates are no longer tied to a single location. The main features of global software development are geographically decentralised organisations (Korkala, Abrahamsson 2007) and global outsourced software development (Schmidt, Meures 2016). Benefits of global software development come from, distributed organisations and outsourcing. When the organisation is distributed in several if not to all continents that brings flexibility to the working hours, as the work can be done almost around the clock. As regards to outsourcing, it frees the corporates to concentrate on their core business.

Geographically decentralised organisations use *Distributed Software Development*, where systems and applications are developed at multiple decentralised sites (Ågerfalk, Fitzgerald et al. 2005). Nowadays large corporates are scattered across cities within same countries and abroad. Consequently it is understandable that face-to-face meetings are rear. Geographically decentralised organisations need to use other methods for communication and collaboration, such as video conferencing, instant messaging and screen sharing tools.

Communication and collaboration supporting distributed software development are effected by three distance dimensions (Ågerfalk, Fitzgerald et al. 2005), (Ågerfalk, Fitzgerald 2006), (Holmstrom, Conchuir et al. 2006) temporal distance, geographical distance and socio-cultural distance.

Temporal distance means that people cannot meet at the same point in time. This is because the work is done in different time zones or in different shifts. If people from Europe, Asia and America work together they may not have any common working hours in a day.

Geographical distance is measured by the difficulty of moving from one location to another. The difficulty of the traveling has more effects to the experience of distance, than only the distance measured in kilometres. Traveling to an office site in the same country, which is hard to get to, can be more difficult than traveling to on office near an airport in another country.

Socio-cultural distance refers to how difficult it is to understand another person's values and practices. The factors of socio-cultural distance are organisational culture, national culture, and language. Within a single country the organisational culture of different companies may vary a great deal, while the corporate culture of a global company may be the same on different continents.

2.7 Communication and collaboration in agile teams and ARTs

Communication and collaboration are essential parts of agile software development (Sharp, Helen, Robinson 2010), (Uludag, Kleehaus et al. 2019.), (Kropp, Meier 2016). Agile team members collaborate through many activities, including planning iterations, testing solutions, solving problems, and estimation, implementation, and development of stories (Sharp, Helen, Robinson 2008). Agile methods strive to support communication and collaboration through several of its practices (Kropp, Meier 2016).

However, agile was developed to be used in small individual teams and it is most effective in co-located teams that sit together in open office environment (Uludag, Kleehaus et al. 2019.). This brings a challenge, as the software development in large corporations is often globally distributed, but dependent on several interdependent teams working together toward common goal (Inayat, Salim et al. 2017), (Uludag, Kleehaus et al. 2019.). Because of this, there is a demand for collaboration and communication technologies and practices, which support both globally distributed software development and the increasing need for scaling agile (Calefato, Ebert 2019), (Sharp, Helen, Robinson 2008).

Agile manifesto (Beck, K., Beedle et al. 2001) emphasises collaboration and communication in many of its values and principles (Robinson, Sharp 2010), (Calefato, Ebert 2019). Agile values talk about focusing more on interaction between people and collaboration with the customer, rather than concentrating on processes, tools, and negotiations.

"Individuals and interactions over processes and tools "

"Customer collaboration over contract negotiation"

Agile principles place the focus on the people working together no matter the role and prioritise face-to-face conversation as the best way to transfer information.

"Business people and developers must work together daily throughout the project."

"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation".

Agile methods introduce several collaborative activities such as iteration planning, daily stand-up, retrospective and iteration review. These regular iteration events foster collaboration and communication in the agile teams (Kropp, Meier 2016). Purpose of a daily stand-up as an internal meeting of the team is to support information flow inside the team by sharing information of current status of tasks, and to improve awareness of others progress in the team. Daily stand-up meeting also enhances collaboration when team members help each other in problem solving and sparring around tasks that are currently in progress in the team (Inayat, Salim et al. 2017), (Sharp, Helen, Robinson 2010). In the same way the other iteration meetings enhance collaboration and communication inside the team. This happens through sharing knowledge and working together on planning, reviewing, and inspecting the iterations (Kropp, Meier 2016).

XP also introduces communicative and collaborative artifacts including wall, story cards, pair programming and planning game. The agile team gathers around the wall during the daily stand-up to discuss what has been done since last meeting and what will be done before the next one. The wall shows the progress of the team through the story cards that the team is working on, making it a key communication and collaboration element for the team (Sharp, Helen, Robinson 2010). Wall and the story cards work well in a large room environment with a co-located team (Sharp, Helen, Robinson 2008).

A good example of collaborative agile artifact is a pair programming, where the program code is developed by two coders sitting together beside one screen. The pair has a common will to enhance actively collaboration and to contribute to it during the coding work (Sharp, Helen, Robinson 2010).

The planning game (Sharp, Helen, Robinson 2010) is held in the beginning of an iteration and also during it. In there the agile team and customer make together a plan for the next iteration. Game is played by customer setting and communicating the prioritisation

for the stories and the team giving an estimation for them. Next the participants negotiate on, which of the stories are moved to the iteration backlog.

Transparency has been recognised as one of the benefits of SAFe methodology (Putta, Paasivaara et al. 2018a), which is also one of the core values of SAFe. SAFe increased transparency in cross team dependencies and communication. Other benefits recorded include a) having better management and detection of dependencies, b) improved collaboration and synchronisation between teams, and c) enhanced alignment, visibility, and communication across globally distributed organisation (Ebert, Paasivaara 2017).

SAFe utilises communication and collaboration practices for agile methods and present new ones including PI Planning to support communication and collaboration in scaled agile initiatives. In team level teams collaborate and share information through scrum events such as daily-stand-up, iteration planning and other iteration events (Paasivaara, M. 2017). To increase communication and collaboration, program level uses scales agile events like scrum of scrum meeting and introduces new meetings such as program increment (PI) planning, system demo, and PO and ART syncs (Razzak, Richardson et al. 2018).

Scrum of scrums event used in scaled agile environment enhances communication between teams as they are forums (Kalenda, Hyna et al. 2018) for a) stimulating discussion, b) coordinating dependencies, and c) bringing transparency to the progress and impediments of the ART. In the same way PO sync[28] offers product owners a forum for information sharing and collaboration and ART sync[29] introduces a platform for scrum masters and product owner to come together in a collaborative manner.

PI planning event offers framework for information flow across teams as it brings all the members of ART together into same physical location (Uludag, Kleehaus et al. 2019.). In PI planning event teams work together solving dependencies and planning the next increment. PI planning event also aims at sharing information and to be transparent of the status of the ART. This is done by presenting vision, mission and architectural runway in beginning of the PI planning event (Uludag, Kleehaus et al. 2019.). Aim of PI planning event is to have established, aligned ART teams, to foster building of social networks and communication channels, and to ease and support collaboration across teams and in the ART (Razzak, Richardson et al. 2018). System demo brings teams

[28] Scaled Agile, Inc, Program Increment. https://www.scaledagileframework.com/program-increment/, [12.1.2020]
[29] Scaled Agile, Inc, Program Increment. https://www.scaledagileframework.com/program-increment/, [12.1.2020]

together to share information on the tasks and statuses of the teams of the ART and to present the achievements of the teams in the current increment.

Scaling agile together with the need for distributed organisations brings a challenge to communication and collaboration in agile teams and agile release trains (Paasivaara, Maria, Durasiewicz et al. 2008), (Razzak, Richardson et al. 2018). Corporations are using nearshore and offshore distributed teams and even members from co-located teams use remote working (Sharp, Helen, Robinson 2008). Artifacts of agile methods, that are used in co-located environment need to be transformed into electronic based communication solutions and collaboration technologies (Sharp, Helen, Robinson 2008), (Calefato, Ebert 2019). It is not an easy task to replace face-to-face interaction in and between teams (Sharp, Helen, Robinson 2008), but there are several communication solution and collaboration tools and technologies that have been developed to support it (Calefato, Ebert 2019), (Sharp, Helen, Robinson 2008), (Kropp, Meier 2016).

Communication technologies can be asynchronous like emails, forums, and wikis or synchronous such as instant messaging and meetings using audio and videoconference tools (Calefato, Ebert 2019). Communication and collaboration platform like Skype for Business (Inayat, Salim et al. 2017), (Paasivaara, M., Durasiewicz et al. 2009) and Microsoft Teams (Calefato, Ebert 2019) support communication and collaboration with the possibilities to chat online, send files and to have a possibility to use audio and video in the meetings. Microsoft teams provide in addition a possibility for several forum walls with different topic and a file storage with simultaneous file collaboration. Online collaboration tools (Kropp, Meier 2016) such as GIT and GitHub, Jenkins, Trello, and JIRA help to simplify the collaboration and they make the progress of the tasks more visible for everyone in the team. GIT is a code version system that works well with distributed teams who need to share code. GitHub repository for codes offers bug tracing and task management. Jenkins tool gives feedback of build and overall statuses to the users. Trello can be used to manage user stories, with building task and scrum boards. JIRA is an issue tracking system and a project tool for managing backlogs (Paasivaara, Maria, Durasiewicz et al. 2008).

As the communication and collaboration in the distributed teams are more challenging than in co-located teams, frequent visiting to the other locations is seen to be a good practice of building and maintaining trust and to enhance collaboration in distributed teams (Paasivaara, M., Durasiewicz et al. 2009). Usually the communication and col-

laboration between people is easier to continue online after seeing at least once each other face-to-face.

To get insights on how communication and collaboration works in distributed agile teams and agile release trains in real- life setting in a large corporation: a case study was performed during 2019. Chapter 3 presents the research objectives, methods, and research design of the case study conducted in 'Company A'.

3 Research objective, methods, and design

Objective of the research was to find out in real-life setting the effects of SAFe methodology on communication and collaboration in teams and agile release trains. To support this objective a case study was conducted in a multinational corporation that is called in this thesis 'Company A' and will remain anonymous in this thesis. A small literature review was made to support the theory of communication and collaboration in agile settings, as well as to make an overview of the agile and SAFe methods, values and practices. Purpose of the case study was to answer three research questions: "How do the practitioners perceive the effects of the events of the SAFe methodology on communication and collaboration?", "Do the ways of working of the SAFe methodology add transparency and value to the practitioners?", and "What are the correlating factors that effect the experience of the practitioners of the SAFe methodology?".

The case study consists of two data gathering methods: a web-survey and interviews. It was conducted in autumn 2019 and provided a large amount of data. In the chapter 3. the data analysis will be presented and the results of the data analysis is looked into in the Chapter 4.

3.1 Research strategy

Case study as a research method is used when one strives to get a deep understanding of a contemporary phenomenon and when the research objective is investigated in real-life context (Farquhar 2012, pp. 3-14). Case study is a snapshot of contemporary phenomenon in a single point in time (Jensen, Rodgers 2001), (Eriksson, Koistinen 2014, pp. 22-44). Case study is a preferred method when 1) seeking answers to questions: when, how and why, 2) researcher has only some control over events, and 3) when interest is in a contemporary phenomenon (Farquhar 2012, pp. 3-14), (Eriks-

son, Koistinen 2014, pp. 22-44), (Yin 2009, pp. 9-15). In case study research it is common to use several different data sources and methods in parallel. The objective is to understand the contemporary phenomenon, not to explain it (Eriksson, Koistinen 2014, pp. 11-21).

The purpose of case study survey (Durepos, Mills et al. 2010, pp. 125-126) is to ask questions from a sample of a group in interest and to find out respondent's opinion, belief or knowledge on a contemporary phenomenon that researcher is investigating. The survey can be conducted only once, which results as a sample of a contemporary phenomenon at one point in time. Information gained in the survey is validated only by belief to the respondents' honesty and willingness to participate. Surveys are also the best way to guarantee anonymity to the participants. They can contain closed form questionnaires with only certain responses allowed or open form questionnaires where responses can be anything.

Interviews (Durepos, Mills et al. 2010, pp. 496-499) are a commonly used data collection method, which allows making systematic inquiry into respondents' knowledge on the topic of interest. An interview can be seen to be a process that by means of conversation can be used as a method to gain an understanding and knowledge of a contemporary phenomenon under investigation. There are many ways of conducting an interview including in person, by phone, online and by email. Interviews are usually audio- or video recorded and the recordings are transcript. Transcriptions are then used as a data source for analysis. A semistructured interview includes pretermined questions, but there is also room for ad-hoc questions that emerge in the situation. With semistructured interviews the researcher's aims at making comparison between the interviewees' answers and to understand deeper their own experience. Semistructured interviews are effective when researcher has some subject matter expertise and the researcher is interested in widening their own understanding of it.

Data analysis (Eriksson, Koistinen 2014, pp. 22-44), (Farquhar 2012, pp. 84-99) is maybe the most difficult phase in a case study. Purpose of the case study is to find answers to its research questions and objectives, and to offer meaning, explanation, and understanding on the phenomenon under investigation. Data analysis can be started with a preliminary analysis, which is more about exploring the data, than making a deep analysis of it. A preliminary analysis helps the researcher to get familiar with the data and to get started with finding patterns and relationships in it. A quantitative data analysis is

about what kind of measurable information the data provides, for example a number of respondents that have the same opinion on a question. With quantitative data analysis frequencies, patterns, tendency and dispersion of the data can be found. Qualitative data analysis has general methods to analyse data by giving them classification, categorisations, and themes. It can begin in a very early stage of the data gathering, for example making preliminary analyses of the semistructured interview by making short notes soon after conducting it. In all research the interpretation of the data is in a crucial position. An interpretation of data can be done in two ways: through direct interpretation or coding of data. The direct interpretation relies on the researcher to have sufficient researcher experience, to be erudite, and to have an overall view to the topic. Data coding is a tool to manage data, to capture the research objectives, and to interpret the data. Coding can be hierarchic with levels to narrow the findings to smaller groups.

3.2 Case study as a research strategy in 'Company A'

Case study was chosen as the research strategy as it was most suited for the objectives of the thesis. The purpose of the case study was to gain deeper understanding of a contemporary phenomenon and to investigate it in a real-life context. The contemporary phenomenon in this thesis case was the usage of agile and SAFe methods in 'Company A' and its effects on communication and collaboration in 'Company A's' teams and organisations. The case study was made as a snapshot, a view to a contemporary phenomenon at a single point of time. The intent was to get answers to specific questions by investigating events, and searching for explanations and interpretations of data.

The theory of the use of SAFe applied in the case study came from two sources: a literature review and the thesis writer's own experience with SAFe methodology. The literature review was done by a) studying scientific articles concerning communication and collaboration in agile and SAFe settings and b) reviewing the purpose of SAFe methodology found on SAFe home-page, SAFe training material, and whitepapers written about SAFe. The thesis writer's own knowledge and experience on the topic was based on previous research made about SAFe for studies, official SAFe trainings attended, and working with agile and SAFe methods for some years.

With the analysis of the data the aim was to find possible patterns and to match them with the theory (Eriksson, Koistinen 2014, pp. 22-44). In this case study there were two kinds of sources of material used in parallel: a web-survey and interviews. A prelimi-

nary analysis of the web-survey data was made using the statistical data available in real-time in survey tool and by preparing preliminary reports in Power BI tool to get some overall comprehension of the possible patterns of the data. After each interview the audio recording was listened and notes were made. Coding of data was used in a loose meaning of it. The notes were marked with code and timestamps of sections corresponding to interviewee discussing interesting topics in respect to the objectives of the case study and were the topics were supporting the findings of the preliminary analysis of the web-survey. The analysing method of the notes was more intuitive than coded, making the interpretation of the data a direct interpretation (Eriksson, Koistinen 2014, pp. 22-44). Direct interpretation approach suited this thesis, as the author had pre-knowledge of the 'Company A', its organisations, way of working, projects, and the way of using agile and SAFe.

3.3 Research design of the case study in the 'Company A'

The case study was conducted during autumn 2019 in 'Company A' as a part of master thesis for computer science. It was conducted in two phases as a web-survey and interviews. The purpose of the case study was to investigate whether events and ceremonies of the Scaled Agile Framework (SAFe) methodology effect communication and collaboration in ARTs and agile teams in them. Among the events daily stand-up meeting, program increment (PI) planning, and system demo were the most interesting items for this case study. The focus of the web-survey and the interviews was in these three events, with additional questions on iteration events, SAFe training, agile coach, and ways of working. To construct the baseline for respondents, questions were asked about a) respondents background, b) which events are held in their ART, and c) what events the respondents attend to. The aim was also to understand how ceremonies and events are held; in the way they are intended or in some other manner. In the end of each question group, there were questions posed to determine if the respondent felt the ceremony in focus, had an effect on information flow, transparency, and collaboration. The respondents were also asked to give ratings from 1 to 5 about the events.

3.3.1 Company A

The case study was conducted anonymously in a multinational corporation, which is in this thesis is called 'Company A'. 'Company A' has its business units and IT organisa-

tion distributed in several countries and has outsourced parts of its IT work into different continent. ‘Company A’ uses vendors that provide both business and IT services, with consultants who work all over the wold. As many distributed corporations, ‘Company A’ utilises distributed software development leading to that agile teams and agile release trains are highly distributed. The teams and ARTs are effected by all three distance dimensions: temporal distance, geographical distance and sociocultural distance (Ågerfalk, Fitzgerald et al. 2005). Remote working is supported and it is very popular in ‘Company A’. That brings a situation where even employees working in same office premises are not present simultaneously all the time.

‘Company A’ started to implement SAFe methodology in mid-2010. ‘Company A’ has SAFe’s solution trains and agile release trains running in several business and IT organisations. Some of the business and IT organisations use a team level scrum, but also the waterfall way of working is still in use. This means that the organisations often work in different syncs. The asynchronousity brings challenges when there are dependencies between big initiatives in different organisations. Some synchronisation between initiatives, projects, and ARTs has been started, and it will increase in the future, when more and more units start using SAFe methodology.

In ‘Company A’ there are many of organisations, solution trains, tens of ARTs, and hundreds of teams using the SAFe methodology, which sums up to thousands of employees. Solution Trains consist from two to five ARTs, ART sizes ranging from 100 to 150 people and teams from 5 to 15 team members. This means that a considerable amount of communication and collaboration is needed to get these big initiatives toward a common goal.

‘Company A’ is using English as the corporate language. English is a native language hardly to any of the employees. Not using one’s native language in the work also brings challenges to the communication and collaboration, resulting deficient understanding.

3.3.2 Timetable and data gathering

Preparations for case study in ‘Company A’ started in March 2019 with a planning of web-survey questions. That included the decision on question groups, formulation of questions per group and their options for answer. In summer 2019, the actual survey form was created with Microsoft Forms tool. Distribution of the web-survey to responders was done in the beginning of September 2019 and the deadline for giving respon-

ses was in the end of September 2019. During October 2019 the interview questions were made and the interviews were held in November 2019. The interview questions were based on the questions of the web-survey with a purpose of getting deeper understanding of the phenomenon.

3.3.3 Web-survey

In the survey phase of the case study the web-survey made with Microsoft Forms was published. The survey contained seven question groups with sub-groups and all in all 74 questions. It included pre-determined questions with answer options, rating questions, and free text questions. For the pre-determined questions the respondents needed to choose their answer among pre-determined options. In addition to the rating questions there was free text space, where the respondent could clarify the reason for giving a weak rating for a particular rating question. In the end of the survey, there was a space for free text, to give any kind of comments the respondent wanted to.

The total number for questions that were intended for gathering information about background, attending SAFe trainings, and working with agile coach was inclusive and increased the total number of questions. The reason for having such an inclusive background check was well intended, as it made it possible to make deeper analyses and find patterns for the possible reasons for answering the questions. In the survey there were eleven question groups. They are described in Table 3.1.

Question group	**Description**
Background	Purpose of *Background* was to get background information on respondent's 1) role and seniority in the role, 2) area of work: in IT or Business, 3) overall seniority as a employee, 4) seniority in working with agile and SAFe, and 5) in what setup and SAFe level the respondent work in.
SAFe training	Purpose of *SAFe training* was to get answers to questions on has the respondent 1) attendance in SAFe trainings, 2) which ones, 3) when, 4) did they find it useful, and 5) if not, what was missing.
Agile coach	Purpose of *Agile coach* was to get answers to questions on the respondent's 1) working with agile coach, 2) on what level the coaching was, 3) was the agile coach local, and 4) did they find the agile coach useful.
Team ceremonies	Purpose of *Team ceremonies* was to find out what team ceremonies are used in respondent's team and about their content. The question group 'Team ceremonies' had two sub-categories: 'Daily stand-up meeting' and 'Iteration events'.

Daily stand-up meeting	Purpose of *Daily stand-up meeting* was to find out 1) how often the meeting is held, 2) size of the team, 3) are the three question: what has been done, what will be done, and is there any impediments on the way, asked, 4) what the content of the meeting is, 5) is the meet up used instead of solving problems in the meeting, 6) does the meeting increase information flow, 7) does the meeting increase collaboration in the team, 8) rating of the meeting, and 9) if they rated weak, why.
Iteration event	Purpose of *Iteration event* was to find out 1) content in each of the event, 2) if the events increase information flow, 3) do the events increase collaboration in the team, 4) rating of the events, and 5) if they rated weak, why.
Agile Release Train (ART) ceremonies	Purpose of *Agile Release Train (ART) ceremonies* was to find out what ART ceremonies are used in respondent's ART and about their content. The question group Agile Release Train (ART) ceremonies' had two sub-categories: 'Program increment (PI) planning' and 'System demo'.
Program Increment (PI) planning	Purpose of *Program Increment (PI) planning* was to find out 1) how the preparations for the event are done, 2) what the content of the event is, 3) who the respondents meet face-to-face and work with in the event, 4) are dependencies between teams discussed, 5) is the mission and vision shared in the event, 6) is the overall knowledge of the status increased, 7) does the meeting increase information flow in the team and between the teams, 8) does the meeting increase collaboration in the team and between the teams, 9) rating of the meeting, and 10) if they rated weak, why.
System demo	Purpose of *System demo* was to find out 1) is the system demo held in each iteration and does the respondent attend it, 2) does the team prepare for system demo together, 3) do most of the teams have demonstration in the system demo, 4) how do teams present progress, 5) does it give transparency on what is happening in other teams, 6) do the events increase collaboration in the team, 7) rating of the events, and 8) if they rated weak, why.
Way of Working (WoW)	Purpose of *Way of Working* was to get answers to questions 1) what way of working the respondent's ART uses, 2) do they add value, and 3) do they increase transparency.
Overall experience	Purpose of *Overall experience* was to get answers to questions on 1) how does the respondent rate overall experience on using SAFe, 2) if they rated weak, why, and 3) if respondent has any kind of comments.

Table 3.1: Web-survey question groups.

The distribution of the link to the web-survey was done via e-mail, directly to individual respondents known to be working with SAFe, to team leaders to be forwarded to their teams, and to RTEs of ARTs. The Web-survey was replied by 65 respondents, representing eleven roles. Two of responses were excluded from analysis, as they did not fulfil pre-requirement of having worked with SAFe methodology. This was recognised from the respondent's own written comment.

Web-survey was made with Microsoft Forms tool, which is a tool for collecting infor-

mation with surveys, quizzes and polls (Microsoft Corporation 2019). It provides users with a questionnaire part and an analysis part. In ‘Questions tab’ the user can build question groups and questions. The questions answer types can be chosen from a radio button, a check box, text, date, and rating. In ‘Responses tab’ user can find ready-made analyses about the answers in form of diagrams. The results can be exported to Ms Excel, Power BI or other tools for more detailed analysis. The whole survey can be found in attachment 1.

3.3.4 Interviews

The interview phase included four interviews and they were held in October 2019. Each interview was conducted with a representative from a different agile / SAFe role, and they represent four different projects and ARTs in the ‘Company A’. Roles chosen were 1) IT analyst representing the team member point of view, 2) scrum master representing the team facilitation point of view, 3) product owner representing the customer point of view, and 4) agile coach representing the activities facilitator point of view. The interviewees were carefully chosen, the prerequisite being that they could well represent the point of view of their role. The criteria for selecting the interviewees were a) they were very well acquainted to the role and SAFe methodology and b) that they had good vision and experience into agile and SAFe way of working. In addition to enable as broad view as possible, the interviewees all represented different projects and ARTs. If interviewees would have come from same project or ART, they would only represent the way of working of one project or ART.

The interview questions based on the web-survey questions in order to deepen the understanding gained from the survey. In addition, there were questions based on the interviewee’s role. Additional questions were asked of SoS, PO and ART sync events where interviewee attends as a representative of the team. Interviews were held as semistructured interviews, which allowed asking ad-hoc questions that arised during the conversation with the interviewee. All interviews were recorded with Skype for Business tool. Summarised transcriptions were made after the interviews and basic level transcriptions were made from the parts used in the thesis.

Interviews were conducted so that the interviewees had a possibility to talk freely. Questions were guiding to the topics, but with space to let them choose what they wanted to bring up on the topic in question. Information on interviews can be seen in Table

3.2. and the interview questions can be found in attachment 2.

Interview	Role	Duration	Meeting style	Point of view
Interview 1	IT analyst	1 h 09 min 51 sec	face-to-face	Team member role
Interview 2	Scrum master (SM)	1h 05 min 12 sec	face-to-face	Facilitator role
Interview 3	Product owner (PO)	1 h 05 min 02 sec	via Skype	Customer role
Interview 4	Agile coach	1 h 19 min 36 sec	face-to-face	Coach activities

Table 3.2: Information on the interviews.

3.3.5 Analysis methods of the case study

During the analysis process of the case study some tools and methods were used to support going through the collected data. The web-survey and interviews produced a large amount of data that needed to be analysed, defined and classified. Data could be combined in several ways depending on the point of view taken. In order to determine what factors influenced on the outcome, many tables and diagrams were produced to find the possible correlation between the factors. All in all there were tens of tables and support tables made, tens of diagrams drawn, and twenty-four pages of transcriptions written based on the interviews.

Tables and diagrams were made by using Power BI and Microsoft Excel tools. The tools complemented each other. Even though Power BI is an efficient tool for crunching data and easy to use for creation of tables and diagrams, it also has its short comings. The tables and diagrams were drawn straight from the fixed data fields and there were no modification and combining possibilities of the data. Microsoft Excel was used to modify, combine and refine the data.

For transcription the transcripts were written with Microsoft Word and the recording of the interviews were listened with Windows media player. During the transcription of the interviews from an audio record, a Live Transcribe -app was used for real-time transcription of speech from the interviews. Live Transcribe -app supported capturing the answers from the interviewees. Simple coding with timestamps was used for the documenting of the transcriptions. This was to make it easier and faster to find the topic of interest from the transcription pages.

Finally, in chapter 4 the most interesting topics among the analyses were chosen to be presented as a result of the case study. The result presents findings from the analysis of

the three key events in focus daily stand-up meeting, PI planning event, and system demo, with findings from iteration events. Chapter 4 also introduces additional findings including results from SAFe training, ways of working, and agile coach.

4 Result

This case study was conducted to find answers to the question "Does the SAFe methodology offer communication mechanisms and collaboration tools to teams and agile release trains" in real-life setting. The scientific literature and their case studies support this hypothesis in several occasions. The literature also brings up the challenges that distributed organisations create to the software development. The result from the case study of this thesis conducted in 'Company A' supports other case studies done in the software development field. Furthermore, the result confirmed what the scientific articles suggested about the communicative and collaborative characteristics of agile and SAFe methods in teams and ARTs.

In this case study the focus was on the increment and iteration events used in SAFe methodology. Particular interest was on the three major events: daily stand-up, PI planning, and system demo. Questions to be answered were: a) "How do the practitioners perceive the effects of the events of the SAFe methodology on communication and collaboration?", b) "Do the ways of working of the SAFe methodology add transparency and value to the practitioners?", and c) "What are the correlating factors that effect the experience of the practitioners of the SAFe methodology?".

The purpose of the case study was also to uncover and understand the underlining factors behind the scenes that correlated and effected the participants' answers. Factors like location of team members, accuracy of following the intended content of the event, and fluency on conducting the events. The findings of this case study supported the hypotheses presented in the literature that agile and scaled agile way of working has a positive effect on communication and collaboration. The result is introduced through the findings from the analysis of the three main events of SAFe, with some additional key findings from the case study.

4.1 Daily stand-up meeting

Daily stand-up meeting was chosen as an event of interest, as it has a big impact on all

the team members. Daily stand-up meeting is held every or almost every weekday. The daily-stand up meeting is an important event to share information, discuss dependencies, and solve impediments in the team.

The case study showed that in 'Company A' the daily stand-up meeting was the most commonly held iteration event and it was held in most of the teams every weekday. Usually the daily stand-up meetings are held in 'Company A' via communication platforms Skype for Business or Microsoft teams.

In most of the teams the three questions "What have you done since the last meeting?", "What do you plan to do before the next meeting?" and "Are there any impediments in your way?" where asked. However the last question was asked more seldom than the two others. From the interviews few things emerged. First of all, in the mature agile teams the questions are not anymore asked as the purpose and the agenda is clear for the team members and all the team members offer and share the information automatically. As the second, in larger teams with over ten members the 15 minutes is too short a time to go over all the three questions with every member. That might have an impact on leaving sometimes the last question out, as there is a feel of haste in the meeting.

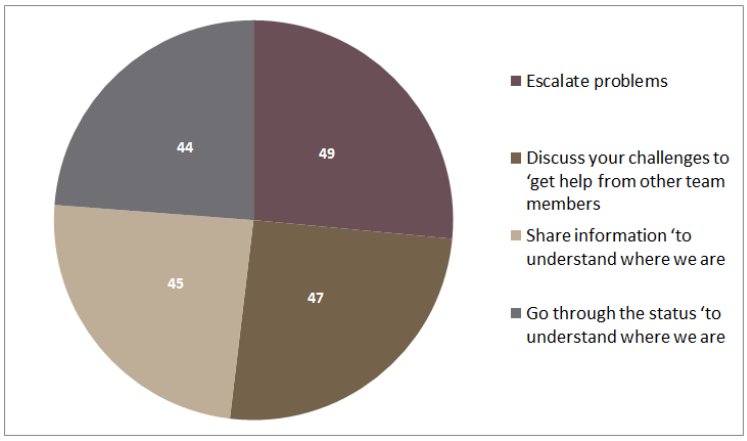


Figure 4.1: Topics in the Daily stand-up meeting.

The question no. 26. "In your Daily stand-up meeting you" was asked to understand how many of the topics that are essential in the daily stand-up meetings realise in the meetings in 'Company A'. From 59 respondents that answered the questions, it could be seen that in most of the cases the purpose of the meeting was fulfilled well. "Escalating problems" emerged as the most frequent topic of discussion in the respondent's team (see Figure 4.1). "Escalating problems" and "Discuss your challenges to get help from other team members", were seen among the key topics in the daily stand-up meeting also by the interviewees.

Sharing information was also found as one of the most important topics in the daily stand-up meeting, according to the interviewees. Sharing information for example about what the members have heard from other ARTs that can effect the work of the team, was felt very important.

The purpose of meet after meeting is to guide discussions emerged in the daily stand-up meeting to be discussed after the meeting. Discussions are held with only relevant team members present, so saving the time from the rest in the team. According to the respondents meet after meeting is held, at least sometimes, in almost all of 'Company A's' agile teams (question no 27). The interviewees also confirmed that meet afters are held in their teams. The meet after was felt to be a good way to continue the discussion of the topics emerged in the daily stand-up meeting. Meet after also helps with the people's busy schedule, as there is no need to call for another meeting for the discussion.

The case study also focused on the effect the daily stand-up meetings have to the communication and collaboration of the team, as this was the one of top interests in the study. In the survey questions about the respondents' view on the effect daily stand-up meeting have on collaboration and communication was asked with two questions addressing information flow and collaboration between the team members (questions no. 28. and 29.).

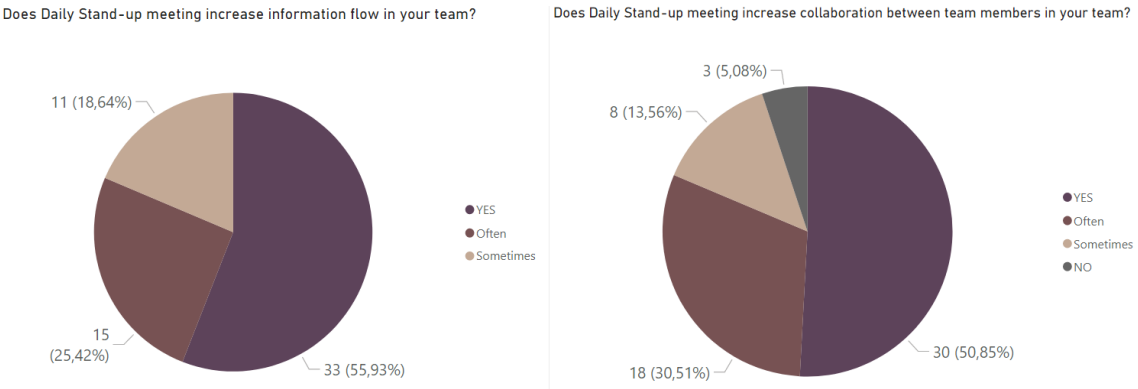


Figure 4.2: Information flow and collaboration in the Daily stand-up meeting.

The results indicate that the respondents view was principally positive as regards to the effect of the daily stand-up meeting on the information flow and collaboration in the team. With both information flow and collaboration, 81% of respondents felt the daily stand-up meeting, at least often, to have an increasing effect on them (see figure 4.2).

When looking at factors that could have had an effect on the answers, few reasons were found. One of them emerged from an interview with a scrum master. The interviewed

scrum master has an agile team that is mostly co-located. All team members except one sit together in a same location. In that case the interviewee did not feel that the daily stand-up meeting increased the information flow and the collaboration in the team significantly. But the interviewee still pointed out two exceptions 1) the effect can be found on those days that employees work remote and 2) the daily stand-up was essential for interaction with the team member, who is not co-located with rest of the team.

"*Some people work sometimes remotely, then having a daily stand-up meeting has a bigger effect on the communication and collaboration. Plus the one person in India would be very out of the loop, if we would not have the daily stand-up meeting*." Scrum master.

The results indicate the positive effect was felt the most, by the respondents in a distributed agile team (question no. 37.), and when the daily stand-up meeting was held every weekday (question no. 21). In these cases the percentage of the positive effect on information flow increased to 88 % and collaboration to 91%. This supports the theory of agile events introducing communication and collaboration mechanisms into distributed teams. The interviewees also confirmed this when question 'In your opinion how does daily stand-up meeting effect collaboration and communication in the team?' was asked.

" *I would say that the effect is big on communication and collaboration, as this is our only common meeting where everyone is together*." Product owner

"*When you are working in different locations, this is one of the few times when you hear how things are going with others*." IT analyst

"*I would say that it helps a lot, especially those teams that are not in the same location*." Agile coach

The most positive high level effect of daily stand-up meeting on information flow and collaboration came when all the three questions were asked in the meeting (questions no. 23, 24, and 25). Due to daily stand-up meeting information flow was seen to increase at least often, by 89% of the respondents and collaboration by 95% of the respondents. That indicates that going through the status and impediments in the daily stand-up meeting is felt to support information flow and improve collaboration in the agile teams.

The overall rating (question no. 30.) of daily stand-up meeting was given as average of

3,3 out of 5 (see Table 4.2). The distribution of the rating among respondents can be seen in Table 4.1. When looking into teams who have daily stand-up meeting every weekday and in the meeting the three questions are asked, the highest average rating at 3,6 is reached. Good rating also correlates with the perception that the daily stand-up meeting increases the information flow and collaboration in the team. When respondent gave average rating of at least 3 the information flow rose into 88% and the collaboration into 90%.

How would you rate your Daily Stand-up meeting?1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent	Quantity
5	2
4	21
3	27
2	7
1	2
Total	**59**

Table 4.1: Daily stand-up meeting rating distribution.

Average rating on Daily Stand-up meeting
3,27

Table 4.2: Average rating on Daily stand-up meeting.

Based on the result of the analyses focusing on the daily stand-up meeting, it can be said that the meeting has positive effect on the information flow and collaboration inside agile teams studied in the 'Company A'. The effect is greater in the distributed teams, that have the meeting every weekday and also in the teams that ask the three question. It could be argued, that following the purpose and the agenda intended for daily stand-up meeting, has a positive impact on how the team members perceive the communicative and collaborative character of the meeting.

4.2 Program Increment (PI) planning

Program increment planning event is the most important event in the increment, as its purpose is to align the teams towards the common goal and to be a platform for planning of next increment activities and its dependencies recognition. PI planning event also provides a framework for meeting people face-to-face from different locations, from one's own team and from other teams.

In 'Company A' the PI planning event of agile release train is held approximately ones in every twelve weeks. PI planning events are held in one physical location, usually in one large room. Members of the ART will gather from five European countries for this two-day event. Offshore ART members participate online, if at all.

The agenda in the PI planning follows the standard agenda given by SAFe methodology

in general, but based on the respondents' answers it could be seen that is not quite so in practice (question no. 36.). The 'Confidence vote' was done in most of the PI planning events, but 'Planning rework' was done only in 15 % of the events (see Figure 4.3).

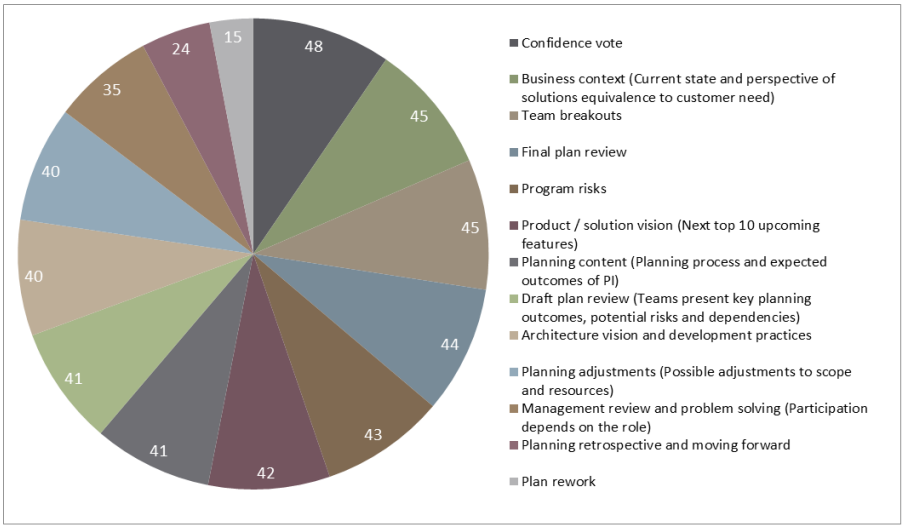


Figure 4.3: Agenda topics usage in PI planning event

An interviewee provided an insight to this, as the interviewee suggested that because planning rework is in the end of the event and having it would mean a longer day for the participants, they vote 3 as they don't want to stay for doing re-work. Second idea suggested by the interviewee was that it could even effect the confidence vote of the team, as with high confidence there is no need to stay and do rework. Other interviewee said about the confidence vote that when it is done openly in the room in front of the whole ART, there is a pressure to vote at least a score 3 out of 5. Voting a score less than 3 means that the team has to give reason for it. Other interviewee's reflection on this was that a need to vote at least score 3, might happen if the team is not very mature and if the spirit inside the ART is not so good. This interviewee believe that in mature teams and ARTs with good spirit, people have the courage to vote also 1 or 2.

One of the focus areas of the case study was the effect of the PI planning event on the communication and collaboration of the team and this was also one of the main interests of the thesis. In the survey, the respondents' view on the effect of the PI planning on communication and collaboration was asked with four questions addressing information flow and collaboration in their own team and between the teams in the ART.

95% of the respondents reported that they have PI planning event in their ART (question no. 32.). The result indicates that the respondents' view was principally positive as regards to effect of the PI planning event on information flow and collaboration in their

own team (questions 44. and 46.). 78 % of the respondents felt PI planning event, at least often, to have an increasing effect on information flow and 71% felt that, at least often, a PI planning event increases collaboration in their own team (see Figure 4.4).

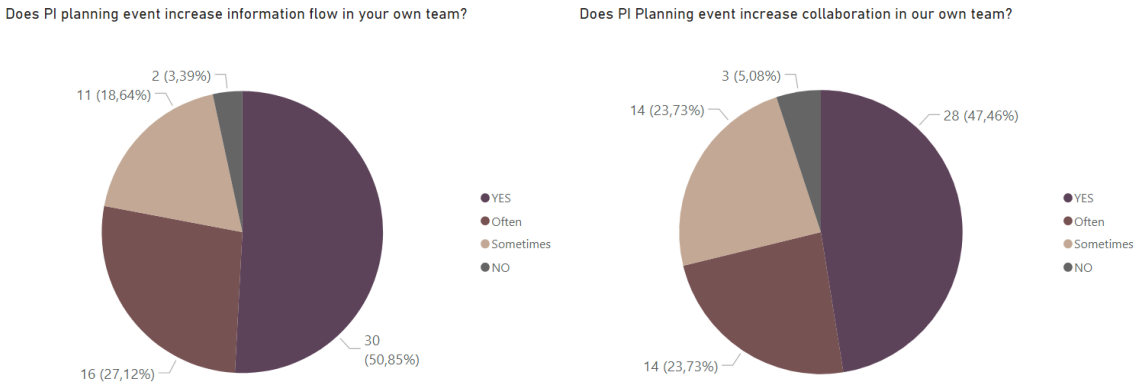


Figure 4.4: Information flow and collaboration inside team in the PI planning event

The positive impact of PI planning on information flow and collaboration inside the team was the highest when members from the distributed teams work together in the event with the team members from their own team (question no. 39.). In this case the impact was felt, at least often, by 89% of the respondents on information flow and by 84% on collaboration. From this it could be concluded, that respondents from distributed teams that meet and work together in PI planning, with their own team members, feel that PI planning event increases communication and collaboration inside the team. An interviewee confirmed that when the team is co-located, the PI planning event does not bring extra communication and collaboration effect for the team. It could also be seen in the analysis that when respondent did not work together with members from one's own team the impact on information flow and collaboration was quite negative from the respondents view.

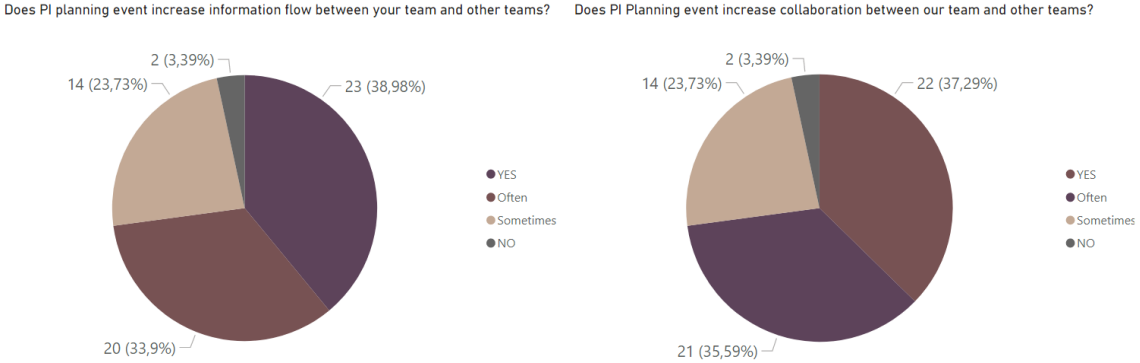


Figure 4.5: Information flow and collaboration between teams in the PI planning event

The results indicate that the respondents' view was quite positive as regards to the effect of the PI planning event on information flow and collaboration between the teams

(questions 45. and 47.). 73 % of the respondents felt PI planning event, at least often, to have an increasing effect on information flow between the teams and 73% felt that, at least often, PI planning event increases collaboration between the teams (see Figure 4.5).

The positive effect of PI planning on information flow and collaboration between teams was felt the highest, when three prerequisites realised in the team a) meet face-to-face members from the other teams (question no. 38), b) work together with team members from other teams (question no. 40), and c) discuss dependencies together with other teams (question no. 41.). In these cases 95% of the respondents felt that PI planning event increases, at least often, information flow and 90% felt that it increases, at least often, collaboration between teams. From this it could be concluded, that respondents from teams that meet, work together, and discuss dependencies in PI planning, feel that PI Planning event increases communication and collaboration between the teams.

Interviewees told that from communication and collaboration perspective of the effects of PI planning, meeting people face-to-face makes it easier to interact also later on.

"*It probably exactly so, that collaboration with teams that you work with, is the most important thing in the PI planning and in a way just to meet people. When you have ones talked to the person or even met with them, the collaboration is easier later.*" Scrum Master

"*PI planning effects communication positively. I have few times said, that it always helps, when you see people face-to-face. It always makes it easier later to have discussions and it is easy to take ad-hoc meetings and chats*." IT analyst

"*I would say that if people are new to each other, meeting each other helps. But that can be the case through any kind of meetings*" Agile Coach

"*It is so, that when we are face-to-face in the same space, it is good to see every ones and awhile. And we do communicate very actively there. So yes I would say it improves it*." Product owner

One communication and collaboration artifact that was brought up in the interviews was the Wall. In the PI planning event the teams have their own boards, and in addition to that the ART has a program board and a dependency board. The boards are physically created to the wall of the room of PI planning event. Boards are used for planning the

stories per iterations covering the whole increment. Working with the wall includes a) making increment plans per team and presenting the plan for all the other teams, b) finding and discussing dependencies between the tasks of the teams, c) keeping the program wall updated, and d) designing dependencies wall so that dependencies are usually illustrated with red strings.

"It is smoother that we just go to the boards of other teams, to see where they have planned specific tasks that we have dependencies with. That way we can see that we can't do dependent tasks of our team before they have finished." Scrum master

The overall rating of PI planning event (question no. 48) was given average of 3,5 out of 5 (see Table 4.4). The distribution of the rating among respondents can be seen in Table 4.3.

How would you rate your PI planning event?1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent	Quantity
5	7
4	23
3	23
2	5
1	1
Total	**59**

Table 4.3: PI planning event rating distribution.

Avarage rating of PI planning meeting
3,51

Table 4.4: Average rating of PI planning event.

The average rating for PI planning was highest at 3,8, when the respondents were members of distributed teams, who a) meet face-to-face, b) work with members from their own and from other teams, and c) discuss dependencies in PI Planning event. Rating also correlates with the perception of does the event increase the information flow and collaboration in the team and between the teams. When respondent gave average rating of, at least 3 1) the information flow rose into 83% and the collaboration into 75% when looking into information flow and collaboration in their own team and 2) the information flow rose into 81% and the collaboration into 79% when looking on information flow and collaboration between the teams.

When the respondents felt that the PI planning event as a whole was well prepared (question no. 35), they thought that after the PI planning event the teams are aligned and have a shared mission and vision (question no. 42). This made also the rating of the PI planning to rise up to 4,4. From the analyses made to the answers, it could be concluded that well prepared PI planning event had the respondent feel that the purpose of the PI planning was fulfilled.

4.3 *System Demo*

System demo provides transparency in the agile release train on what is going on in the other teams, as in system demo all the teams in the ART share their outcomes from the increment. System demo also provides the overall picture on the progress of the ART as a whole.

There are two kinds of demos held in 'Company A' a) iteration demos that are held in the end of each iteration and b) system demos that are held in end of each increment. 90% of the respondents reported that they have a system demo in their ART (question no. 32). In the survey the questions concentrated on the increment level system demo, but in the interviews they both were discussed.

The ways of making presentations (question no. 54) in the system demos in 'Company A' had quite much variation (see Figure 4.6). The system presentation being the most popular and Confluence page the least popular. System presentation usually meant showing parts of the system that the team had been working on. Power point presentations, which were also quite popular, were usually presentations from business side, meaning reports, statistics and business pictures. Code presentations can mean a) having coding window open and showing what happens when entering input data and b) showing how API's, frontend and backend programs work.

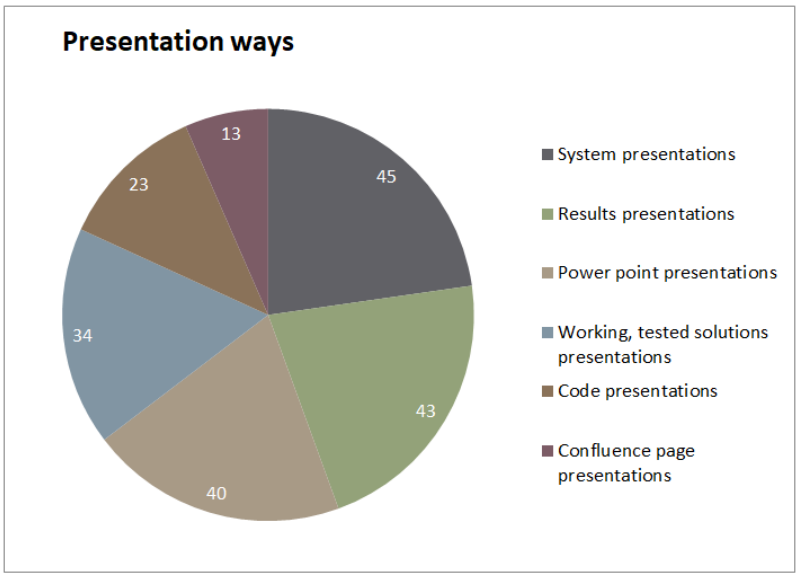


Figure 4.6: Presentation ways in System demo.

As the interest of the case study was in transparency through communication, and collaboration, the respondents and interviewees were also asked about the effect of system demo to them (questions 55. and 56.). As regards to the question 'Does System demo

give transparency to what is happening in the other teams in the ART', 64% of the respondents felt that system demo has, at least often, a positive effect. Respondents and interviewees did not feel that system demo meeting effects the collaboration so much as daily stand-up and PI Planning do. Only 36% of respondents felt, that at least often, the system demo has positive effect on collaboration in the team (see Figure 4.7). Most of the interviewees felt that system demo meeting gives transparency to what is happening in the other teams in the ART.

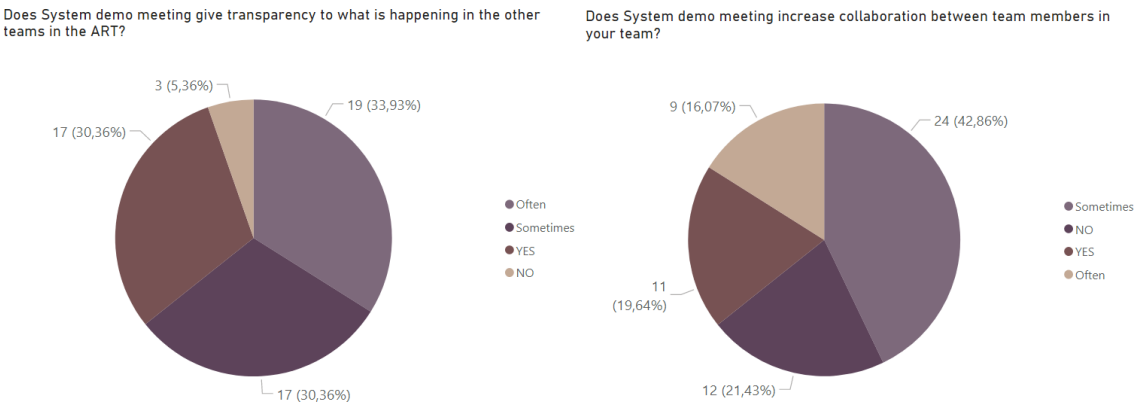


Figure 4.7: Transparency and collaboration in the system demo meeting

The interviewees also felt that system demo does not increase collaboration in the team. To the question 'Does System demo give transparency to what is happening in the other teams in the ART' the interviewees said:

"*Yes, it is nice to see at times what others do."* Scrum master

"Definitely does, you get this kind of overall picture in there." IT analyst

The positive effect was highest among the respondents who answered YES to question 'Does your team prepare for System demo together' (question no. 52.). Then 82% of responders felt that system demo meeting gives transparency to what is happening in the other teams in the ART and 59 % felt that system demo increases collaboration in the team. There were no other explanatory factors found. Responders just don't feel that the system demo session enhanced collaboration in the team very much. The transparency factor was at 64 % also indicating that transparency was not felt by everyone.

An interesting fact emerged from the analysis. Even that the respondents did not find system demo to give transparency to what is going on in other teams very high or that it increases collaboration in the team so much, they still gave a good rating (question no. 57.) of 3,13 out of 5 to the system demo (see Tables 4.6). The distribution of the rating among respondents can be seen in Table 4.5.

How would you rate your System demo?1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent	Quantity
5	3
4	12
3	31
2	9
1	1
Total	**56**

Table 4.5: System demo rating distribution.

Average rating of System demo
3,13

Table 4.6: Average rating of system demo.

The good rating shows that respondents still felt system demo to be a relevant event. Rating also correlated with the perception of does the event give transparency and increase collaboration in the team. When respondent gave an average rating of, at least 3, the transparency rose to 70% and the collaboration to 41%.

4.4 *Iteration events*

As discovered earlier, the daily stand-up meeting was held in almost all of the teams in 'Company A' according to respondents and interviewees, making it the most commonly held iteration event in 'Company A'. All the iteration events were held in 68% of the responders' teams (question no. 20.). Iteration planning was the second most commonly held event and innovation and planning iteration being the least held event (see Figure 4.8). Two of the interviewees told that all iteration events are held in their teams and two told that often backlog refinement event is not held.

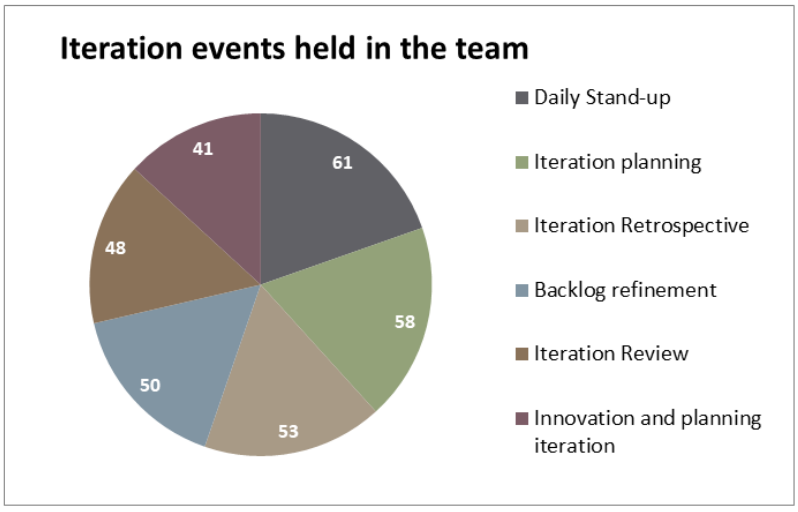


Figure 4.8:Iteration events held in the team

Iteration *planning* events are held in almost all of the teams. The two-thirds of the respondents reported to have tasks (question no. 59.), 'estimation of stories' and 'commitment to delivery amount and iteration goals' included in their iteration planning event. 21% of the respondents answered that in the iteration planning event to support

the estimation work they use card deck when playing estimation poker.

Most commonly performed task of Iteration *Retrospective* event (question no. 62.) was 'identify ways to improve' with 80% of the respondents' teams doing it and 'identify issues' was done in 72% of the teams. The purpose of the retrospective event is to go through what went well and identify areas for improvement, so the answers support doing this in the teams. Task 'Measure performance' was however done in only 25% of the respondent's teams. This is not ideal, as to have a quantitative review of performance; team should gather some kind of metrics. Metrics could be as simple as answering 'yes' or 'no' to a question 'Iteration goals are met in our team'.

Backlog refinement event was held in 79% of the survey respondents' teams. 'Refine stories' task was done in 76% of the teams and 'estimate stories and enablers' task was done in 66% of the teams (question no. 61.). One thing that came up in the analysis was that 'Establish initial acceptance criteria for stories' were done only in 40% of the teams. The situation is not recommendable, as the 'establishing acceptance criteria for stories' task is critical for the declaration of 'definition of done'. If a user stories acceptance criteria is not clear how can a) tests pass, b) code be reviewed, and c) functional test pass? Two of the interviewees were also concerned about the lack of the backlog refinement work.

"Lately we have quite often left out this backlog refinement and that is not a good thing. When we don't have these refinements then the definition of stories in JIRA remain incomplete" IT Analyst

"Most often backlog refinement is missing. Backlog refinement is just the thing, with what the common understanding is built between the team and the product owner or between the team and business" Agile Coach

Iteration *review* event is held in 79% of the respondents' teams. In iteration review event 71% of the respondents' teams discuss user stories that are not completed and the reason why they are not (question no. 60.). In the iteration review event 52% of the teams go through risks and impediments related to the user stories.

When looking into all of the respondents who have some of the iteration events held in their team, also the ones that say they don't have iteration review, it can be seen that only 41 % of 'Company A's' teams discuss risks and impediments in their teams. This doesn't go so well with the agile methods, as their purpose is to optimise predictability,

to control risk and to remove impediments. “Is there something getting in your way?” was also the less asked question in daily stand-up meetings. This is surprising, as also the interviewees felt that escalating problems and solving together impediments was very important to go through in the meeting. From this it could be concluded, that risks and impediments is something that is easily overlooked even that it shouldn’t be. Leaving risks and impediments unsolved can slow down the progress of the development and cause serious errors in the systems.

Although the PI planning and the system demo events of *Innovation and planning* – iteration are held in more than 90% of the respondents teams, the innovation and planning -iteration itself as a two-week-event was not so popular. Innovation and planning -iteration was the least held event of all the iteration events (question no. 63.). It was held, at least often, in 32% and sometimes in 33% of the respondents’ teams. When having the innovation and planning –iteration, 65% of respondents report that they work with technical infrastructure and that only 15% work on final acceptance test and documentation (question no. 64.). Of all the respondents 38% answered that they in innovation and planning -iteration ‘have time for innovation and exploration’ and 13% answered that they have time for ‘educate to support continuous learning and improvement’. So it could be concluded that, the innovation and planning –iteration is something that there seems not to be time left from normal development work in the ARTs. It would be recommendable to dedicate time for innovation, exploration and continuous learning in the agile release train.

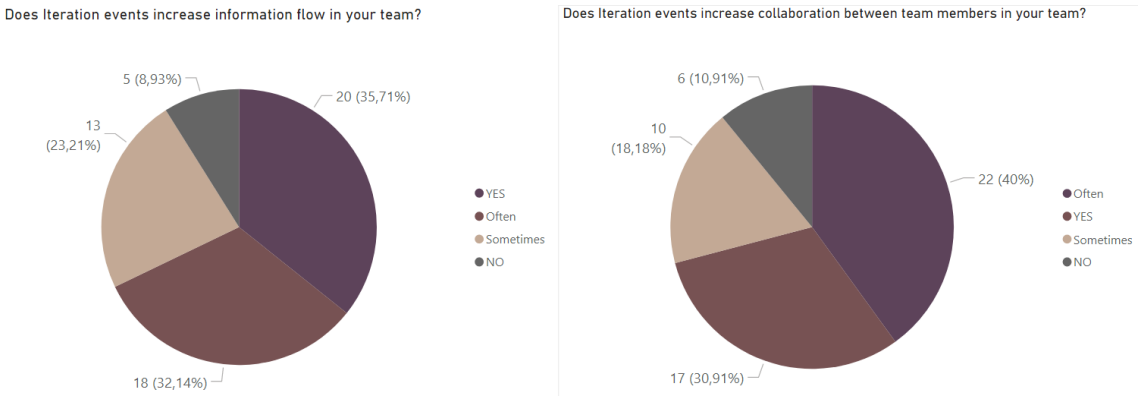


Figure 4.9: Information flow and collaboration in the Iteration events.

The experiences were positive as the regards to the effects of the iteration events on information flow and collaboration in the teams (question no. 65. and 66.), but not so positive, as with the daily stand-up meeting. 68 % of the respondents felt iteration

planning events, at least often, to have an increasing effect on information flow and 71% felt that, at least often, iteration events increase collaboration in their own team (see Figure 4.9).

Iteration events were experienced to be in a good level (question no. 67.), as the average rating was 3,08 (see Table 4.8). The distribution of the rating among respondents can be seen in Table 4.7.

How would you rate your Iteration events?1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent	Quantity
5	2
4	18
3	31
2	7
1	5
Total	**63**

Table 4.7: Iteration events rating.

Average rating of Iteration events
3,08

Table 4.8: Average rating of iteration events.

When the respondents gave a rating of, at least 3, to iteration events, the positive effect raised quite much, information flow up to 82% and collaboration up to 84%. Hence, it could be concluded, that when the iteration events were experienced good quality wise, the effect on information flow and collaboration also rose.

4.5 Additional findings

There were a lot of interesting aspects that could be found in the analysis of the case study: many points of view that could be taken, and many answers for different kind of questions. In this section there are presented some of the most interesting additional findings related to the, SAFe trainings, ways of working, and having an agile coach. The additional findings presents in the end the overall experiences of SAFe and comments from the interviewees on that.

The Annual State of Agile -survey gathers information on the 'Top five Tips for success' when scaling agile. In the case study several of these were also items of interest. The web-survey and the interviews included also questions related to a) having company provided training programs, b) consistent practices and processes across teams, c) implementation of a common tool across teams, and d) internal agile coaches. Next some of the findings about these items are presented.

Training is one of the top five success factors in scaling agile according to the Annual State of Agile survey. In 'Company A' 93% has attended a SAFe training among the web-survey respondents and interviewees (question no. 9.). Among the survey respon-

dents that attended the training, 91% found it to be useful (question no. 12.), (see Figure 4.10). Most of the respondents have had the training in last 2 years (question no. 11.). All the interviewees felt the training was useful and that with the help of the training they gained better understanding through examples, about SAFe terminology, such terms like 'funnel' and 'work in progress'. The interviewees also felt that training gave deeper understanding of the SAFe framework and that it woke up the curiosity to know more about it.

Figure 4.10: Usefulness of the SAFe trainings.

To a question about the usefulness of training an interviewee answered:

"*Definitely I feel that a person that goes into this kind of SAFe training has better conditions to get inside the way of working in projects".* IT Analyst

In 'Company A' the level of SAFe trained practitioners is quite high and according to SAFe methodology and findings in the literature that supports possibilities of being successful in implementing and using SAFe. Respondents and interviewees experience of the usefulness of the trainings also fits to the common perception that training is useful for understanding of SAFe and training helps to work with it.

Of the *ways of working* JIRA was the most used tool in 'Company A' (question no. 69.), as it is used in 96% of the respondents' and interviewees teams. Confluence as a team workspace -tool was also quite common with usage of 81%. Team backlog, either scrum or Kanban was used in 89% of respondent's teams. Features and stories are used in 90% and epic –level in 67% of the teams (See Figure 4.11).

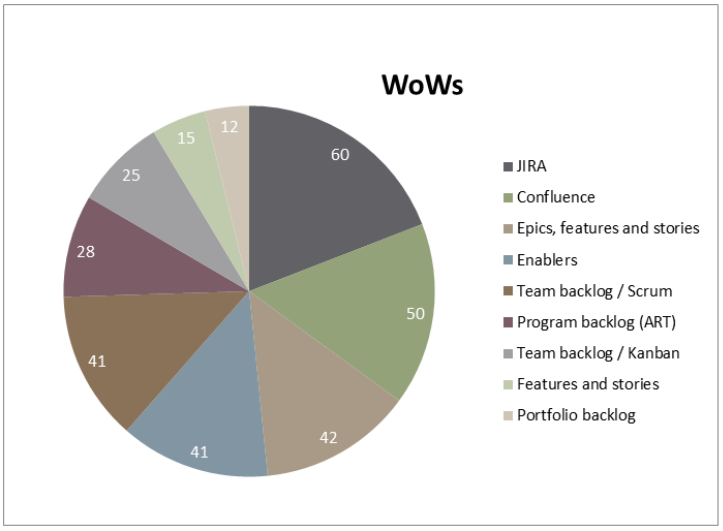


Figure 4.11: Ways of working

In regards to the responders' experiences on way of working, 73% of the respondents felt that, at least often, way of working increased transparency (question no. 71.) and 81% felt that, at least often, way of working adds value for them (question no. 70.) (see Figure 4.12).

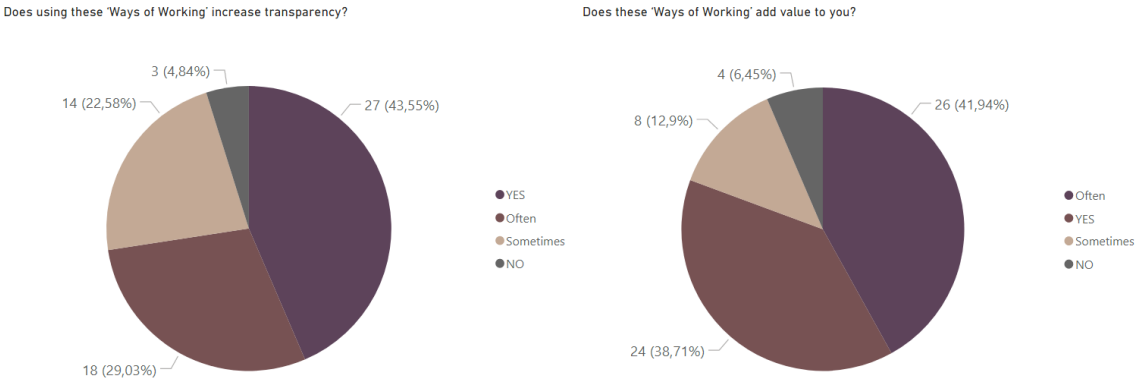


Figure 4.12: Transparency and value in ways of working

In distributed teams the effect of way of working was experienced more positive as 81% felt that the way of working, at least often, increases transparency and 88% felt that, at least often, the way of working adds value to them. Ratings of events also correlate with the perception of does the way of working increase transparency and give value. When respondent gave average rating at least 3 to the SAFe events (questions no. 30, 48, 57, and 67.) where these way of working are used, transparency rose to 78% and the way of working adds value to 87%.

The case study analysis showed that JIRA was the most commonly used project management tool in 'Company A', as it was also reported to be in other companies who answered the 'Annual state of agile survey'. In modern day free-seating and distributed

team environment, it is not always possible to have a physical 'Wall'. Collaboration tools like JIRA offer possibility to build virtual backlog for everyone to see, so increasing transparency and knowledge about the tasks of the team. It can be concluded, that JIRA along with other ways of working was viewed to increase transparency in 'Company A'. That is consistent to what is reported in literature about JIRA making the tasks of the team more visible for the team members.

Currently 53% of the respondents in 'Company A' have an *agile coach* (question no. 14.) and 69% has had one in some point in time (question no. 15.). Among the respondents that have or have had an agile coach 68% of them found that having agile coach to support the team or the ART was useful (question no. 18.), (see Figure 4.13).

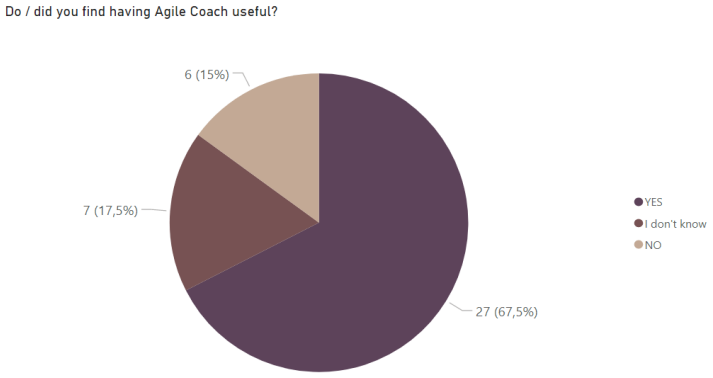


Figure 4.13: Usefulness of the agile coach

86% of the respondents who currently has a co-located agile coach (question no. 17.) in the team or in the ART, found having agile coach useful and 87% of them would like to work more with an agile coach (question no. 19.). Among the respondents who have never have had an agile coach 50% of them would like to work with one (questions no. 14, 15, 19.). Interviewees felt agile coaches useful, as they had facilitated PI planning events and as agile coaches was felt to support the idea of continues learning.

Overall experience of SAFe was rated at 3,35 (question no. 72.), (see Table 4.10). Interesting was that no-one rated overall experience to be weak. The distribution of the rating among respondents can be seen in Table 4.9.

How would you rate your overall experience on using SAFe methodology?1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent	Quantity
5	5
4	22
3	26
2	10
Total	**63**

Table 4.9: Distribution of rating on overall experience of using SAFe.

Average rating of overall experience on using SAFe methodology
3,35

Table 4.10: Average in overall experience on using SAFe.

Interviewees were very positive with the overall experience of SAFe. Survey respondents answered in free text that they found SAFe to a) bring teams together to solve problems, b) add transparency, and c) improve communication and knowledge sharing.

"What is best on my mind in the agile, is its idea that very different people in the team, solve problems fast in real-time and from different points of view. Agile method demands absurd self-driven capabilities, drive and pro-activity. The transparency, it is in fact a matter that I like". Product owner

"In benefits side I would start with overall communication. Our communication and knowledge sharing has improved. The fact that we do plan together, we commit to the goals by doing it together and committing to them together. We aim to share knowledge, to be open, and to work together". Agile coach

"*I would say on how SAFe brings benefits, it brings a lot of them and it brings transparency*". IT analyst

It could be said that the additional findings from 'Company A' support the 'Top five Tips for success' when scaling agile, that is introduced in the Annual State of Agile survey. The respondents and interviewees found a) SAFe trainings to be useful, b) ways of working with common tools to add value, and c) agile coach to be useful and that they would like to work more with them. The respondents and interviewees also felt that the overall experience of the SAFe methodology is good.

4.6 Summary of the result

- The daily stand-up meeting was the most commonly used agile event as it was also reported in the 'annual state of agile survey'. The usage of other iteration events was also following the same popularity order than the 'annual state of agile survey' reported.
- Daily stand-up was experienced by the respondents to increase information flow and collaboration the most, of all the analysed events.
- Within all the rated events in the web-survey the highest rating at 3,5 was given to the program (PI) increment planning event. This was a good result, as the PI planning is said to be the most important event of SAFe by SAFe methodology and the literature in the field.
- System demo was surprisingly not seen as so informative event as it was

expected to be when starting the case study. System demo also got the lowest percentage in collaboration at 36%.

- It could be seen in the analysis that when respondents experienced that iteration events are in good quality, then also the effect on information flow and collaboration was felt more positive.
- What was found lacking in iteration events was going through risks and impediments. That is quite against agile practises and leaving risks and impediments unsolved can lead to delays in the development and cause serious errors in the systems.
- Additional findings include a) almost all of the respondents have been in a SAFe training and felt it useful, b) approximately 61% has worked with agile coach and 68% of them found it useful, and c) 73% of the respondents felt that, at least often, the way of working of the SAFe methodology increased transparency and 81% felt that, at least often, the way of working of the SAFe methodology adds value for them.
- All rated items in the web-survey were given an average above three. This indicates that the respondents experienced, that as a whole, the events and the way of working of SAFe methodology are in a good level in 'Company A'.
- From the results of the analysis, it was clear, that in many cases distributed teams valued more the SAFe events of the team and ART. They felt that the events gave more opportunities for communication and collaboration. Also members of distributed teams felt that the way of working gave more transparency and value for them.

Among the results from the events in focus and from the additional findings some suggestions for improvement and conclusions emerged. The suggestions for improvement are presented with the reflections of the conclusions in chapter 5.

5 Conclusion

Globalisation and distributing the organisation is very common these days in large corporations. Not only manufacturing plants are moved to inexpensive locations, but also software development. This brings the phenomenon of distributed organisations to many

software development practitioners' working life. At the same time agile way of working is becoming more and more popular as a method for software development. Agile methods base on face-to-face communication and collaboration of a co-located team. Because of this, there has been a need to develop communication mechanisms and collaboration tools, to support and simulate 'on location like' work environment for distributed teams and organisations.

This thesis focused on studying the effects of agile and scaled agile methods on collaboration and communication in teams and agile release trains. The study was conducted as an anonymous case study in a multinational corporation. Corporation was in this case study simply called 'Company A'. Scaled agile methodology that is used in 'Company A' is Scaled Agile Framework (SAFe). Therefore this thesis focused exclusively on one scaled agile methodology, SAFe. The case study used two research methods: a web-survey and interviews. Web-survey inquired from respondents about events used in their agile release trains, how the respondents perceived SAFe events, and does agile and SAFe way of working have an effect on communication and collaboration. Interviews focused on the same questions as the web-survey. The aim of conducting interviews was to obtain more dept and insight to the topics addressed in the web-survey.

The result of the case study confirmed quite well the hypotheses that agile and scaled agile methods have positive effect and support team members working life. There were some events that did not have such a big impact, but in general respondents and interviewees felt that agile and scaled agile methods effect positively on communication and collaboration. All the average ratings of events were estimated good. From the analysis of the findings, one thing merged as a contributing factor; distributed teams felt more positive of the effectiveness of the SAFe events and the way of working on communication and collaboration in the teams and ARTs.

The three main events analysed were daily stand-up meeting, program increment (PI) planning, and system demo, along with other iteration events. In addition, SAFe training, agile coach, ways of working and overall experience of SAFe were researched. There were interesting findings found from all of them.

Among the team level iteration events a daily stand-up meeting was the most commonly held one. This supports the findings from other reports in the field that have been made about the usage of agile methods. Daily stand-up meeting was also felt to be an event

that increased the most the information flow and collaboration in the team.

When looking in to the average rating of the events the highest rating of 3,5 out of 5, was given to program increment (PI) planning event. Overall, all the events used in SAFe methodology where experienced to be conducted in a good level. However, it was surprising that system demo was not felt to be as informative as expected and the collaborative effect of it was experienced quite low.

As regards to additional findings ‘Company A‘ has trained almost all of its respondents into SAFe methodology and most of them felt the training to be useful. More than half of the respondents have worked with an agile coach and experienced it to be useful. Most of the respondents found the ways of working of SAFe methodology to add value for them, at least often and that the way of working increased transparency.

One pattern was clearly emerging from the result of the case study; distributed teams felt that they benefited more from the events and way of working. When looking into the results given by respondents in distributed teams; the effects of events and way of working on information flow, collaboration, transparency and value adding was almost always positive. The simple explanation can be that co-located teams communicate and collaborate on more ad-hoc bases and they don’t have so much need for the communication in the events and they don’t have the same need for usage of collaboration tools. But when respondents are not sharing a physical working space with their team members the need for reoccurring, common events and tools, for sharing information and collaboration are more appreciated.

The other pattern that could be seen was that when the event was perceived well organised (average rating at least 3) the events and way of working were experienced to be more informative, transparent and collaborative. The rose of the positive effect could be seen in all the measured events, however highest increase of 13%, could be seen in iteration events. It is quite natural that the quality of the events effects the perceiving of the other aspects of the events.

In the daily stand-up meeting evaluation, it could be seen, that when using the practices that are meant to be used in the meeting, the meeting was estimated to be better in quality. Practices included having the daily stand-up meeting every weekday and asking three questions about status and impediments. In these cases respondents also felt that because of the daily stand-up meeting the information flow and the collaboration

increased in their team.

The case study set out to find answers to three research questions: “How do the practitioners perceive the effects of the events of the SAFe methodology on communication and collaboration?”, “Do the ways of working of the SAFe methodology add transparency and value to the practitioners?”, and “What are the correlating factors that effect the experience of the practitioners of the SAFe methodology?”. Hypothesis was that when used in proper way the methods and way of working have positive effect on communication and collaboration in agile teams and ARTs. The objective of the case study was well reached. The web-survey resulted in an adequate number of responds and the questions asked served the purpose planned for them. Interviews supported the objective to deepen the understanding obtained from web-survey and provided more inside to the usage of SAFe in ‘Company A’. From the findings of the case study, answers to the research questions could be found and some explaining patterns were discovered.

From the result of the case study data some discussion topics for recommendation emerged. Discussion topics like: ‘Would there be ways to make the agile and scaled agile events even more successful by following the attended content of the meetings better?’ and ‘Would putting even more effort on facilitation and preparation of the events increase the appreciation of them?’, ‘Could time spent together in PI planning event be used totally for collaboration?’, ‘Are there ways to dedicate more time for innovation and continuous learning’, and ‘How to put more focus on going through risks and impediments’.

Analysis showed that the daily stand-up meeting got the highest scores among the respondents when the three questions of status and impediments were asked and the meeting was held every weekday. Suggestion for improvement would be to take three questions to be a part of all of the daily stand-up meetings, as well as having the meeting held every weekday.

It was surprising that only the two-thirds of the respondents felt system demo to give them opportunity to learn what is happening in other teams in the ART. This is one of the questions that would require follow up in order to understand the reason behind. Could the reason be that a) the system demos are too long or boring?, b) the system demos are not facilitated as well as they could be?, c) whether the participants of the

demo do not understand each other e.g. business vs. IT terminology, and d) are people just not so interested what the other teams do? It would be interesting to find out what items would give more value for the participants, to be able to give recommendations on how to improve the appreciation of the system demo.

Some of the agile release trains in 'Company A' have the first part of PI planning event held in a separate online meeting a day before. This came up in the interviews and was felt to be a good practice. Recommendation would be to consider taking this as a broader practice. Moving the monologue of presentations of the PI planning Day 1 to be held online, would release more time for teams to work together in their own teams and with other teams. This way the whole time together could be used on planning the increments tasks, discussing dependencies, solving challenges and reducing risks.

The data reveals that the innovation and planning -iteration was not used as much as it could be. Recommendation going forward would be to dedicate time for innovation, exploration, and continuous learning in the programs increment. There are benefits to be gained also from innovative work, alongside the normal development work; it increases motivation, supports learning and allows new idea to emerge.

There was indication that in iteration events going through risks and impediments was not done so much. Agile practises promote quite heavily focusing also on risks and impediments. As leaving risks and impediments unsolved can lead to delays in the development and cause serious errors in the systems. Recommendation is to find ways to promote doing this in the teams and agile release trains.

Although there is every year more research done on scaled agile methodologies, there is still not much research done on, how the scaled agile methodologies support large initiatives in distributed organisations. Topics of interest would be: can scaled agile methods like SAFe, take the place of an old fashioned project management Waterfall method? Can SAFe methodology handle all aspect of project management of large projects including budgeting and governing work of hundreds of project members?

It would have been great to have observation period of the key events as a part of this research, but unfortunately with the timeframe of this thesis it was not possible. Proposal for future research within this topic would be to do real-life observation of scaled agile level events PI planning and system demo. There are also questions that would need to have answers, about the reasons behind the experiences. Including why the sys-

tem demo was not felt to give transparency to what is happening in the other teams and why system demo does not increase collaboration in the their own team.

Communication and collaboration are corner stones of all organisations; this case study took a small look into the real-life situation of using SAFe and its effects on communication and collaboration. For future research, it would be interesting to make this kind of case study in several companies and to compare the results for similarities and possible patterns.

In the end it can be said that this was an educational and interesting journey. Making the case study showed that with a (small) effort a lot can be discovered and that the real-life research is fascinating as it brings dept and reality to the theory from the literature. It also left a hunger for doing more research.

References

ÅGERFALK, P.J. and FITZGERALD, B., 2006. Session details: Flexible and distributed software processes: old petunias in new bowls? *Communications of the ACM*, .

ÅGERFALK, P.J., FITZGERALD, B., HOLMSTROM OLSSON, H., LINGS, B., LUNDELL, B. and CONCHUIR, E., 2005. A framework for considering opportunities and threats in distributed software development.

AIRA, A., 2012. *Toimiva yhteistyö : työelämän vuorovaikutussuhteet, tiimit ja verkostot*, University of Jyväskylä.

ALQUDAH MASHAL and RAZALI ROZILAWATI, 2016. A Review of Scaling Agile Methods in Large Software Development. *International Journal on Advanced Science, Engineering and Information Technology*, **6**, pp. 828-837.

BECK, K., 1999. Embracing change with extreme programming. *Computer*, **32**(10), pp. 70-77.

BECK, K., BEEDLE, A., BENNEKUM, A., COCBURN, W., CUNNINGHAM, M., FOWLER, J., GRENNING, J., HIGHSMITH, A., HUNT, R., JEFFRIES, J., KERN, B., MARICK, R.C., MARTIN, S., MELLOR, K., SCHWABER, J., SUTHERLAND, J. and THOMAS, D., 2001-last update, Manifesto for Agile Software Development2020].

BECK, K., 2000. *Extreme programming explained: embrace change.* addison-wesley professional.

BLANKENSHIP, J., BUSSA, M., FOGGON, D., LEWIS, R. and MILLETT, S., 2011. *Pro Agile .NET development with Scrum.* New York: Apress.

CALEFATO, F. and EBERT, C., 2019. Agile Collaboration for Distributed Teams [Software Technology]. *IEEE Software*, **36**(1), pp. 72-78.

DIKERT, K., PAASIVAARA, M. and LASSENIUS, C., 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, **119**, pp. 87-108.

DUREPOS, G., MILLS, A.J. and WIEBE, E., 2010. *Encyclopedia of case study research.* Los Angeles, Calif. ; London: SAGE.

ERIKSSON, P. and KOISTINEN, K., 2014. *Monenlainen tapaustutkimus.* Helsinki: Kuluttajatutkimuskeskus.

FARQUHAR, J.D., 2012. *Case study research for business.* Los Angeles, Calif. ; London: SAGE.

GROSS, J.M. and MCINNIS, K.R., 2003. *Kanban made simple : demystifying and applying Toyota's legendary manufacturing process.* New York: AMACOM.

GUSTAVSSON, T., 2018 Impacts on team performance in large-scale agile software development, *2018 Joint of the 17th Business Informatics Research Short Papers, Workshops and Doctoral Consortium, BIR-WS 2018, 24 September 2018 through 26 September 2018* 2018, CEUR-WS, pp. 421-431.

HOLMSTROM, H., CONCHUIR, E.O., AGERFALK, P.J. and FITZGERALD, B., 2006. Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance. *2006 IEEE International Conference on Global Software Engineering (ICGSE'06),* , pp. 3-11.

HUMMEL, M., ROSENKRANZ, C. and HOLTEN, R., 2016. The Role of Shared Understanding in Distributed Scrum Development: An Empirical Analysis.

INAYAT, I., SALIM, S.S. and MARCZAK, S., 2017. Survey of communication and awareness as the most relevant socio-technical aspects of requirements-driven collaboration among software development teams. *IET Software,* **11**(6), pp. 277-285.

JENSEN, J.L. and RODGERS, R., 2001. Cumulating the Intellectual Gold of Case Study Research. *Public administration review,* **61**(2), pp. 235-246.

KALENDA, M., HYNA, P. and ROSSI, B., 2018. Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process,* **30**(10), pp. e1954.

KORKALA, M. and ABRAHAMSSON, P., 2007Communication in distributed agile development: A case study, *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007)* 2007, IEEE, pp. 203-210.

KROPP, M. and MEIER, A., 2016. Collaboration and human factors in software development: Teaching agile methodologies based on industrial insight. *2016 IEEE Global Engineering Education Conference (EDUCON),* , pp. 1003-1011.

KUPPUSWAMI, S., VIVEKANANDAN, K., RAMASWAMY, P. and RODRIGUES, P., 2003. The effects of individual XP practices on software development effort. *ACM SIGSOFT Software Engineering Notes,* **28**(6), pp. 6.

LEOPOLD, K. and KALTENECKER, S., 2015. *Kanban and change leadership : creating a cluture of continuous improvement.* Hoboken, New Jersey: Wiley.

LITTLEJOHN, S.W. and FOSS, K.A., 2009. *Encyclopedia of Communication Theory.* Sage Publications Inc.

MICROSOFT CORPORATION, 2019-last update, Microsoft Forms - easily create surveys, quizzes and polls. Available: https://forms.office.com/ [Feb 25, 2020].

MISTRIK, I., GRUNDY, J., WHITEHEAD, J. and VAN DER HOEK, A., 2010. *Collaborative software engineering.* Berlin ; London: Springer.

NEVE, J.R., GODBOLE, K. and NEVE, R., 2017. Productivity and process improvement using "Scaled Agiled" approaches: An emphasized analysis. *2017 International Conference on Inventive Computing and Informatics (ICICI),* , pp. 793-798.

PAASIVAARA, M., 2017. Adopting SAFe to Scale Agile in a Globally Distributed Organization. *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE),* , pp. 36-40.

PAASIVAARA, M., DURASIEWICZ, S. and LASSENIUS, C., 2009. Using Scrum in Distributed Agile Development: A Multiple Case Study. *2009 Fourth IEEE International Conference on Global Software Engineering,* , pp. 195-204.

PAASIVAARA, M., DURASIEWICZ, S. and LASSENIUS, C., 2008Distributed agile development: Using scrum in a large project, *2008 IEEE International Conference on Global Software Engineering* 2008, IEEE, pp. 87-95.

POPPENDIECK, M. and POPPENDIECK, T., 2003. *Lean software development : an agile toolkit.* Boston, MA: Addison Wesley.

PRIES, K.H. and QUIGLEY, J.M., 2010. *Scrum project management.* Boca Raton, FL: CRC Press.

PRIES-HEJE, J. and KROHN, M.M., 2017 The safe way to the agile organization, *Proceedings of the XP2017 scientific workshops* 2017, pp. 1-3.

PUTTA, A., PAASIVAARA, M. and LASSENIUS, C., 2018aAdopting scaled agile framework (SAFe) a multivocal literature review, *Proceedings of the 19th International Conference on Agile Software Development: Companion* 2018a, pp. 1-4.

PUTTA, A., PAASIVAARA, M. and LASSENIUS, C., 2018bBenefits and challenges of adopting the scaled agile framework (SAFe): preliminary results from a multivocal literature review, *International Conference on Product-Focused Software Process Improvement* 2018b, Springer, pp. 334-351.

RAHMAN, S.S., MOLLAH, S.A., ANIRBAN, S., RAHMAN, H., RAHMAN, M., HASSAN, M. and SHARIF, H., 2018. OSCRUM: A modified scrum for open source software development. *International Journal of Simulation: Systems, Science and Technology,* **19**(3), pp. 20.1-20.7.

RAZZAK, M.A., RICHARDSON, I., NOLL, J., CANNA C. NIC and BEECHAM, S., 2018. Scaling Agile across the Global Organization: An Early Stage Industrial SAFe Self-Assessment. *2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE),* , pp. 116-125.

RESNICK, S., BJORK, A. and DE LA MAZA, M., 2011. *Professional scrum with Team Foundation Server 2010.* Indianapolis, IN: Wiley Pub.

ROBINSON, H. and SHARP, H., 2010. Collaboration, Communication and Co-ordination in Agile Software Development Practice. *Collaborative Software Engineering,* , pp. 93-108.

SAYER, N. and WILLIAMS, B., 2007. *Lean for dummies.* Hoboken, NJ: Wiley.

SCALED AGILE INC, 2016. *SAFe® 4.0 Introduction, Overview of the Scaled Agile Framework® for Lean Software and Systems Engineering.* Scaled Agile Inc.

SCALED AGILE INC., 2018. *SAFe® 4.6 Introduction Overview of the Scaled Agile Framework® for Lean Enterprises.*

SCALED AGILE, I., 5.12., 2019-last update, Scaled Agile Framework – SAFe for Lean Enterprises. Available: https://v46.scaledagileframework.com/?_ga=2.251796489.1839718725.1586504500-1971911319.1585467332 [Apr 12, 2020].

SCHMIDT, N. and MEURES, C., 2016. "Mind the Gap": An Analysis of Communication in Agile Global Outsourced Software Development Projects. *2016 49th Hawaii International Conference on System Sciences (HICSS),* , pp. 501-510.

SHARP, H., ROBINSON, H., J. SEGAL and D. FURNISS, 2006. The role of story cards and the wall in XP teams: a distributed cognition perspective. *AGILE 2006 (AGILE'06),* , pp. 11 pp.-75.

SHARP, H. and ROBINSON, H., 2010. Three 'C's of Agile Practice: Collaboration, Coordination and Communication. In: T. DINGSØYR, T. DYBÅ and N.B. MOE, eds, *Agile Software Development: Current Research and Future Directions.* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 61-85.

SHARP, H. and ROBINSON, H., 2008. *Collaboration and co-ordination in mature eXtreme programming teams.*

STOJANOV, I., TURETKEN, O. and TRIENEKENS, J.J.M., 2015. A Maturity Model for Scaling Agile Development. *2015 41st Euromicro Conference on Software Engineering and Advanced Applications,* , pp. 446-453.

TABAKA, J., 2006. *Collaboration explained: facilitation skills for software project leaders.* Addison-Wesley.

ULUDAG, Ö, KLEEHAUS, M., ERCELIC, S. and MATTHES, F., 2019.Using Social Network Analysis to Investigate the Collaboration Between Architects and Agile Teams: A Case Study of a Large-Scale Agile Development Program in a German Consumer Electronics Company, *Agile Processes in Software Engineering and Extreme*, Programming 2019., Springer International Publishing, pp. 137-153.

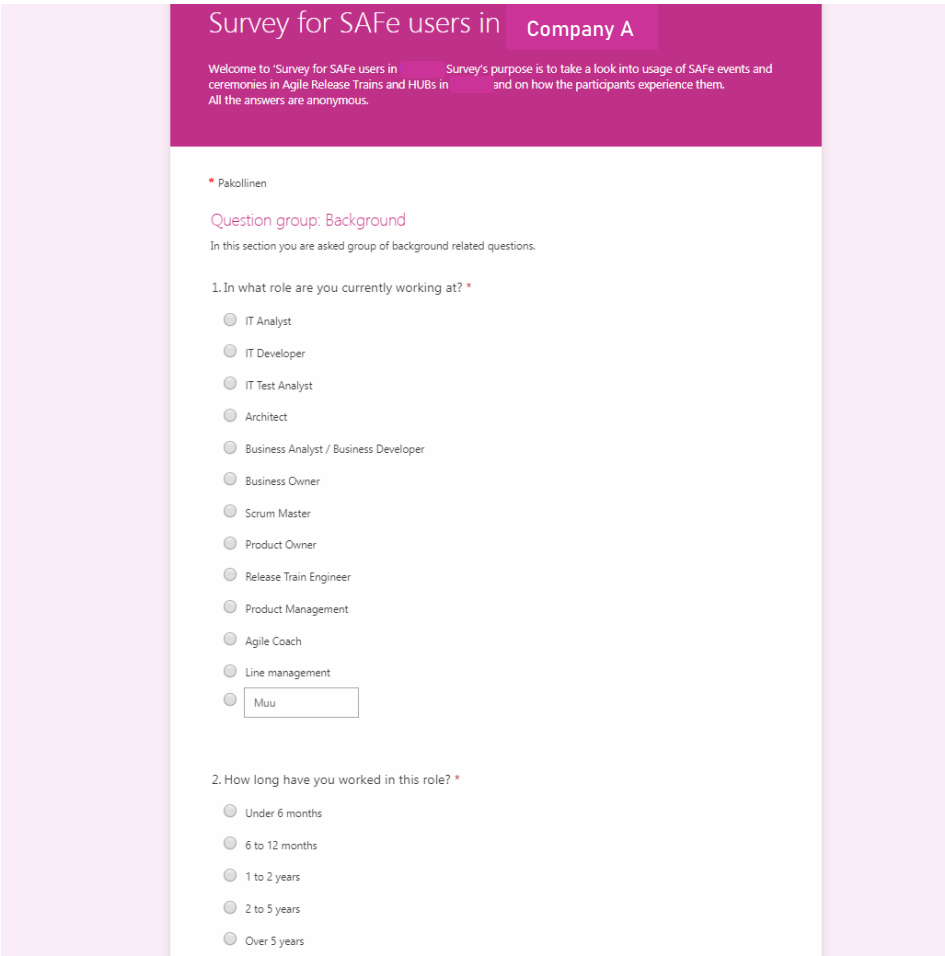
VERSIONONE INC., 2019-last update, 13th Annual State of Agile Survey. Available: https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508 [Feb 23, 2020].

YIN, R.K., 2009. *Case study research : design and methods.* 4th edn. Thousand Oaks (Calif.): Sage Publications.

.

Attachments

Attachment 1. Survey questions



Survey for SAFe users in Company A

Welcome to 'Survey for SAFe users in Survey's purpose is to take a look into usage of SAFe events and ceremonies in Agile Release Trains and HUBs in and on how the participants experience them.
All the answers are anonymous.

* Pakollinen

Question group: Background

In this section you are asked group of background related questions.

1. In what role are you currently working at? *

- IT Analyst
- IT Developer
- IT Test Analyst
- Architect
- Business Analyst / Business Developer
- Business Owner
- Scrum Master
- Product Owner
- Release Train Engineer
- Product Management
- Agile Coach
- Line management
- Muu

2. How long have you worked in this role? *

- Under 6 months
- 6 to 12 months
- 1 to 2 years
- 2 to 5 years
- Over 5 years

3. Where do you work / have you worked *

- IT
- Business
- Both
- Muu

4. How long have you worked in IT / Business or in both? *

- Under 6 months
- 6 to 12 months
- 1 to 2 years
- 2 to 5 years
- Over 5 years
- Over 10 years
- Over 15 years
- Over 20 years

5. How long have you worked agile? *

- Under 6 months
- 6 to 12 months
- 1 to 2 years
- 2 to 5 years
- Over 5 years

6. How long have you worked with SAFe? *

- Under 6 months
- 6 to 12 months
- 1 to 2 years
- 2 to 5 years
- Over 5 years

7. In what kind of execution set-up do you work? *

- ART (Agile Release Train)
- HUB
- Muu

8. What level do you work in? *

- Team level in an ART (Agile Release Train)?
- Program level in an ART
- Portfolio level
- Muu

Seuraava

Question group: SAFe training

In this section you are asked group of SAFe training related questions.

9. Have you attended SAFe training? *

- YES
- NO

10. What SAFe training have you attended? You can choose several.

- SAFe for teams
- Leading SAFe
- SAFe Scrum Master
- SAFe Product Owner / Product Manager
- SAFe Release Train Engineer
- Implementing SAFe
- SAFe DevOps
- Muu

11. When did you attend your latest SAFe training?

- Less than 1 year ago
- Less than 2 years ago
- Less than 3 year ago
- More than 3 years ago

12. Did you find SAFe training useful?

- YES
- NO
- I don't know

13. If you didn't find the training useful, what was missing?

Kirjoita vastaus

Question group: Agile Coach

In this section you are asked group of Agile Coach related questions.

14. Do you have an Agile Coach? *

- ○ YES, our team has
- ○ YES, our ART has
- ○ No coach
- ○ I don't know

15. Have you had an Agile Coach? *

- ○ YES, our team has had
- ○ YES, our ART has had
- ○ No coach
- ○ I don't know

16. Was the coaching happening in? You can choose several.

- ☐ Team level
- ☐ Program/ ART level
- ☐ Individual level
- ☐ No coaching
- ☐ I don't know

17. Is / was the Agile Coach co-located?

- ○ YES
- ○ NO
- ○ I don't know

18. Do / did you find having Agile Coach useful?

- ○ YES
- ○ NO
- ○ I don't know

19. Would you like to work more with an agile coach?

- ○ YES
- ○ NO
- ○ I don't know

Question group: Team ceremonies

In this section you are asked group of team ceremonies related questions. In sections later you will be asked more detailed questions about the team ceremonies.

20. Which meetings are held in your agile team? You can choose several.

- [ ] Iteration planning
- [ ] Daily Stand-up
- [ ] Iteration Review
- [ ] Backlog refinement
- [ ] Iteration Retrospective
- [ ] Muu

Osa 5

Question group: Team ceremonies, Daily Stand-up meeting

In this section you are asked group of Daily Stand-up meeting related questions.
Assumption is that the Daily Stand-up is 15 minutes long.

21. How often do you have Daily Stand-up meeting? *

- ( ) Every week day
- ( ) 3 times a week
- ( ) 2 times a week
- ( ) 1 time a week
- ( ) I'm not in an agile team

22. How many members are there in your agile team?

- ( ) Under 5 team members
- ( ) 5 to 10 team members
- ( ) 10 to 15 team members
- ( ) Over 15 team members

23. "What have you done since the last meeting?" question is asked in your Daily Stand-up?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

24. "What do you plan to do before the next meeting?" question is asked in your Daily Stand-up?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

25. "Is there something getting in your way?" question is asked in your Daily Stand-up?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

26. In your Daily Stand-up meeting you: You can choose several.

- [ ] Go through the status 'to understand where we are '
- [ ] Share information 'to understand where we are '
- [ ] Escalate problems
- [ ] Discuss your challenges to 'get help from other team members'
- [ ] Muu

27. When needed, you continue solving challenges and discuss in 'meet after' following your Daily Stand-up

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

28. In your experience does Daily Stand-up meeting increase information flow in your team?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

29. In your experience does Daily Stand-up meeting increase collaboration between team members in your team?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

30. How would you rate your Daily Stand-up meeting?
1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent

1 2 3 4 5
( ) ( ) ( ) ( ) ( )

31. If you answered weak, why?

Kirjoita vastaus

Osa 6 ...

Question group: Agile Release Train (ART) ceremonies

In this section you are asked group of ART ceremonies related questions. In sections later you will be asked more detailed questions about the ART ceremonies.

32. Which events are held in your Agile Release Train? You can choose several. *

- [ ] Program Increment (PI) planning
- [ ] System Demo
- [ ] Inspect & Adapt
- [ ] Scrum of Scrums (SoS)
- [ ] Product owner (PO) Sync
- [ ] Agile Release Train (ART) Sync
- [ ] Muu

Osa 7 ...

Question group: ART ceremonies, Program Increment (PI) planning

In this section you are asked group of PI planning related questions.

33. From your team to preparation of PI planning event participates: You can choose several. *

- [ ] All team members
- [ ] Some team members
- [ ] Scrum master
- [ ] Product Owner
- [ ] Architect
- [ ] I don't know

34. You do enough preparation work for PI planning event in your team *

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

35. I feel your PI planning event is as a whole well prepared *

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

36. In your Program Increment (PI) planning event you have: You can choose several. *

- [ ] Business context (Current state and perspective of solutions equivalence to customer need)
- [ ] Product / solution vision (Next top 10 upcoming features)
- [ ] Architecture vision and development practices
- [ ] Planning content (Planning process and expected outcomes of PI)
- [ ] Team breakouts
- [ ] Draft plan review (Teams present key planning outcomes, potential risks and dependencies)
- [ ] Management review and problem solving (Participation depends on the role)
- [ ] Planning adjustments (Possible adjustments to scope and resources)
- [ ] Final plan review
- [ ] Program risks
- [ ] Confidence vote
- [ ] Plan rework
- [ ] Planning retrospective and moving forward
- [ ] Muu

37. In PI planning event I meet face-to-face team members from my own team who work in different location *

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

38. In PI planning event I meet face-to-face team members from other teams *

- YES
- Often
- Sometimes
- NO

39. In PI planning event I work together with team members from my team *

- YES
- Often
- Sometimes
- NO

40. In PI planning event I work together with team members from other teams *

- YES
- Often
- Sometimes
- NO

41. In PI planning event we together discuss dependencies between my team's and other team's tasks *

- YES
- Often
- Sometimes
- NO

42. I feel that all teams are aligned and have shared mission and vision after PI meeting *

- YES
- Often
- Sometimes
- NO

43. In your experience does PI planning event increase your overall knowledge of the Agile Release Train's status? *

- YES
- Often
- Sometimes
- NO

44. In your experience does PI planning event increase information flow in your own team? *

- YES
- Often
- Sometimes
- NO

45. In your experience does PI planning event increase information flow between your team and other teams? *

- YES
- Often
- Sometimes
- NO

46. In your experience does PI planning event increase collaboration in your own team? *

- YES
- Often
- Sometimes
- NO

47. In your experience does PI planning event increase collaboration between your team and other teams? *

- YES
- Often
- Sometimes
- NO

48. How would you rate your PI planning event?
1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent *

1	2	3	4	5
○	○	○	○	○

49. If you answered weak, why?

Kirjoita vastaus

Question group: ART ceremonies, System demo

In this section you are asked group of System demo related questions.

50. Does your Agile Release Train have System demo in each of the iterations? *

- YES
- Often
- Sometimes
- NO

51. Do you attend System demo? *

- YES
- Often
- Sometimes
- NO

52. Does your team prepare for System demo together?

- YES
- Often
- Sometimes
- NO

53. Does the System demo usually include demonstrations from most of the teams in the Agile Release Train?

- YES
- Often
- Sometimes
- NO

54. In your System demo teams present their progress with: You can choose several.

- [ ] System presentations
- [ ] Code presentations
- [ ] Results presentations
- [ ] Working, tested solutions presentations
- [ ] Confluence page presentations
- [ ] Power point presentations
- [ ] Muu

55. In your experience does System demo meeting give transparency to what is happening in other teams in the ART?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

56. In your experience does System demo meeting increase collaboration between team members in your team?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

57. How would you rate your System demo?
1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent *

1 2 3 4 5
( ) ( ) ( ) ( ) ( )

58. If you answered weak, why?

Kirjoita vastaus

Question group: Team ceremonies, Iteration events

In this section you are asked group of Iteration event related questions.

59. In your Iteration planning you: You can choose several.

- [ ] Go through feedback from prior iterations, system demo and other teams
- [ ] Establish team capacity
- [ ] Estimate size for stories
- [ ] Estimate size for stories with card deck in a estimation poker or version of it
- [ ] Finalise iteration backlog
- [ ] Make commitment to the delivery amount
- [ ] Make commitment to the Iteration goals
- [ ] Muu

60. In your Iteration Review you: You can choose several.

- [ ] Review business context, Iteration Goals and team PI Objectives
- [ ] Demo and go through feedback of each Story, Enabler, and NFR
- [ ] Discuss Stories that are not completed and why
- [ ] Go through current risks and impediments
- [ ] Revise Team Backlog and team PI Objectives
- [ ] Muu

61. In your Backlog refinement you: You can choose several.

- [ ] Refine stories and enablers
- [ ] Review stories and enablers
- [ ] Estimate stories and enablers
- [ ] Establish initial acceptance criteria for stories
- [ ] Muu

62. In your Iteration Retrospective you: You can choose several.

- [ ] Discuss the result of the iteration
- [ ] Identify issues
- [ ] Review your practices
- [ ] Identify ways to improve
- [ ] Measure your performance
- [ ] Muu

63. Do you have Innovation and Planning (IP) Iteration?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO
- ( ) I don't know

64. In your Innovation and Planning (IP) Iteration you: You can choose several.

- [ ] Have time for innovation and exploration
- [ ] Work on technical infrastructure, tooling and other impediments
- [ ] Educate to support continuous learning and improvement
- [ ] Dedicate time for PI system demo, I&A workshop, PI planning and backlog refinement
- [ ] Have final integration of the solution, including verification and validation
- [ ] Have final user acceptance test and documentation
- [ ] Muu

65. In your experience do Iteration events increase information flow in your team?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

66. In your experience do Iteration events increase collaboration between team members in your team?

- ( ) YES
- ( ) Often
- ( ) Sometimes
- ( ) NO

67. How would you rate your Iteration events?
1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent *

1	2	3	4	5
( )	( )	( )	( )	( )

68. If you answered weak, why?

Kirjoita vastaus

Question group: Way of working (WoW)

In this section you are asked group of WoW related questions.

69. In your Agile Release Train/ HUB you use: You can choose several. *

- [ ] Features and stories
- [ ] Epics, features and stories
- [ ] Enablers
- [ ] Confluence
- [ ] JIRA
- [ ] Team backlog / Scrum
- [ ] Team backlog / Kanban
- [ ] Program backlog (ART)
- [ ] Portfolio backlog (HUB)
- [ ] Muu

70. In your experience does these 'Ways of Working' add value to you? *

- ○ YES
- ○ Often
- ○ Sometimes
- ○ NO

71. In your experience does using these 'Ways of Working' increase transparency? *

- ○ YES
- ○ Often
- ○ Sometimes
- ○ NO

Osa 11 ···

Question group: Overall experience

72. How would you rate your overall experience on using SAFe methodology?
1 = Weak, 2 = Fair, 3 = Good, 4 = Very good, 5 = Excellent *

1	2	3	4	5
○	○	○	○	○

73. If you answered weak, why?

Kirjoita vastaus

74. Anything else you want to add? Please feel free to comments.

Kirjoita vastaus

Attachment 2. Interview questions

Interview questions: common and role specific event questions

Theme: Background

1) What is your current role?
2) How long have you worked in:
   a) role
   b) agile
   c) SAFe
3) What trainings in SAFe have you attended?
   a) Did you find the training useful?
   b) Any other feedback on the training?
4) Have you worked with agile coach?
   a) Did you find it useful?
   b) Any other feedback on the coaching?
5) Agile Team
   a) What is your team's size at the moment?
   b) How is your team located? In how many locations?
   c) How do you communicate?
   d) In your opinion does this have effect in your communication and collaboration?
   e) What other roles are there in your team?
   f) What are your responsibilities and tasks?
6) Agile Release Train
   a) How many teams and persons are in your ART?
   b) How long has the ART been running?
   c) In how many locations are the people in your ART?
   d) Is your ART part of a Hub or a Portfolio?
   e) How would you describe the complexity level of your train, for example dependencies to other teams in the train? (high_medium_low)

Theme: Team ceremonies

1. In your agile team do you have?
   a. Iteration planning?
   b. Daily stand-up?
   c. Iteration review?
   d. Backlog refinement?
   e. Iteration Retrospective?

Theme: Team ceremonies, Daily stand-up

1. Can you describe to me how does your typical daily stand-up looks like to you?
   a. How often does your team have it
2. To have a successful daily stand-up, what do you consider to be the most important things and is this realised in your opinion in your team?
3. What is expected from you in these meetings?
4. How do you contribute to these?
5. What works well?
6. What would you change?

7. Is there something stopping you (impediments)?
8. Do you use meet after in your agile team?
9. In your opinion how does Daily stand up effect collaboration and communication?

Theme: Team ceremonies, Iteration events

1) Can you tell me about your team's?
   a) **Iteration planning**?
   b) Do you:
      (a) Go through feedback from prior iterations, system demo and other teams
      (b) Establish team capacity
      (c) Estimate size for stories
      (d) Estimate size for stories with card deck in a estimation poker or version of it
      (e) Select stories
      (f) Finalise iteration backlog
      (g) Make commitment to the delivery amount
      (h) Make commitment to the Iteration goals
   c) **Iteration review**?
   d) Do you:
      (a) Assess progress
      (b) Show working stories to get feedback
      (c) Demo stories
      (d) Go over the iteration goals and discuss their status
      (e) Walk-trough all committed stories
      (f) Reflect on which stories were not completed and why
      (g) Determine how the progress is towards PI objectives
      (h) Refine the team backlog
   e) **Backlog refinement**?
   f) Do you:
      (a) Refine stories and enablers
      (b) Review stories and enablers
      (c) Estimate stories and enablers
      (d) Establish initial acceptance criteria for stories
   g) **Iteration Retrospective**?
   h) Do you:
      (a) Discuss the result of the iteration
      (b) Identify issues
      (c) Review our practices
      (d) Identify ways to improve
      (e) Measure our performance
      (f) Identify issues
2) How often does your team have them, in every iteration?
3) To have a successful iteration event, what do you consider to be the most important things and is this realised in your opinion in your team?
4) What is expected from you in these meetings?
5) How do you contribute to these?
6) What works well?
7) What would you change?
8) Is there something stopping you (impediments)?
9) How do you see that these events effect communication and collaboration in your team?

Theme: ART ceremonies and events

1) Do you have?
   a) Program Increment planning
   b) Innovation and planning iteration
      i) (Includes PI planning, I&A, time for innovation ex.)
         (a) Have time for innovation and exploration
         (b) Work on technical infrastructure, tooling and other impediments
         (c) Educate to support continuous learning and improvement
         (d) Dedicate time for PI system demo, I&A workshop, PI planning and backlog refinement
         (e) Have final integration of the solution, including verification and validation
         (f) Have final user acceptance test and documentation
   c) Inspect & Adapt
      i) (includes PI system Demo, Quantitative measurements , Retrospective and problem solving workshop)
   d) System Demo (if not in I&A)
   e) Scrum of Scrums
   f) Product owner sync
   g) Agile release Train Sync

Theme: ART ceremonies and events, Program Increment (PI) planning

1. Do you participate on preparations for PI Planning? How?
2. Can you describe to me, how does your typical PI planning look like to you?

More specific PI questions

3. What is included in your PI planning event?

Program Increment (PI) planning agenda	
Day 1	Day 2
Business context (Current state and perspective of solutions equivalence to customer need)	Planning adjustments (Possible adjustments to scope and resources)
Product / solution vision (Next top 10 upcoming features)	Team Breakouts
Architecture vision and development practices	Final plan review
Planning context (Planning process and expected outcomes of PI)	Program risks
Team breakouts	Confidence vote
Draft plan review (Teams present key planning outcomes, potential risks and dependencies)	Plan rework
Management review and problem solving (Participation depends on the role)	Planning retrospective and moving forward

4. Is your PI Planning held in one or several locations?
   a. What are the methods and tools used to handle this?
   b. Your view on F2F PI versus having it in different locations?
5. To have a successful PI, what do you consider to be the most important things and are they realised in your opinion?
6. How do you contribute to these?
7. What works well?
8. What would you change?
9. Is there something stopping you (impediments)?

10. Do you feel that all teams are aligned and have shared mission and vision after PI planning meeting?
11. In your opinion does PI planning event increase your overall knowledge of the Agile Release Train status?
12. Do you in PI planning event work together with your team members?
    a. From other locations
13. Do you in PI planning event work together with members from other teams?
14. Do you in PI planning event discuss dependencies between your team's and other team's tasks?
15. If we take a communication and collaboration view to PI planning event, how do you see that having PI planning event effects:
    a. Communication in your own team?
    b. Communication between teams in the ART?
    c. Collaboration in your own team?
    d. Collaboration in between teams in ART?

System demo

1) Can you tell me about your system demo
   a) How often do you have a system demo (in every increment)?
   b) Do all teams from your ART participate?
      i) What do you think about that?
   c) In our System demo how do teams present their progress?
      (a) System presentations
      (b) Code presentations
      (c) Results presentations
      (d) Working, tested solutions presentations
      (e) Confluence page presentations
      (f) Power point presentations
      (g) other
   d) In your opinion does System demo meeting give transparency to what is happening in other teams in the ART?
   e) Do you feel that preparing for system demo increases communication and collaboration in your own team?

ART ceremonies and events

2) Can you tell me about your ART's?
   a) Inspect & Adapt
      i) Includes (PI system Demo, Quantitative measurements and Retrospective and problem solving workshop)
   b) Innovation and planning iteration, do you?
      (a) Have time for innovation and exploration
      (b) Work on technical infrastructure, tooling and other impediments
      (c) Educate to support continuous learning and improvement
      (d) Dedicate time for PI system demo, I&A workshop, PI planning and backlog refinement
      (e) Have final integration of the solution, including verification and validation
      (f) Have final user acceptance test and documentation
   c) How do you see that these events effect communication and collaboration in your train?

Role dependent event questions

d) **Scrum of Scrums**
   i) Can you describe to me how does your typical SoS look to you?
   ii) To have a successful SoS , what do you consider to be the most important things and is this realised in your opinion in your team?
   iii) What is expected from you in these meetings?
   iv) How do you contribute to these?
   v) What works well?
   vi) What would you change?
   vii) Is there something stopping you (impediments)?

e) Product owner sync?
   i) Can you describe to me how does your typical PO sync look to you?
   ii) To have a successful PO sync, what do you consider to be the most important things and is this realised in your opinion in your team?
   iii) What is expected from you in these meetings?
   iv) How do you contribute to these?
   v) What works well?
   vi) What would you change?
   vii) Is there something stopping you (impediments)?
   viii) How do you see that these events effect communication and collaboration in your train?

f) **Agile release Train Sync?**
   i) Can you describe to me how does your typical ART sync look to you?
   ii) To have a successful ART sync, what do you consider to be the most important things and is this realised in your opinion in your team?
   iii) What is expected from you in these meetings?
   iv) How do you contribute to these?
   v) What works well?
   vi) What would you change?
   vii) Is there something stopping you (impediments)?

g) How do you see that these events effect communication and collaboration in your train?

Theme communication

- How would you describe in overall communication in your team and ART?

Theme collaboration

- How would you describe in overall collaboration in your team and ART?

Theme Benefits and challenges

- What benefits you see that SAFe brings into Software Development?
- What challenges you see that SAFe brings into Software Development?

Open word, what would you like to add?